

---

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: N2612 – Elektrotechnika a informatika  
Studijní obor: 3901T025 – Přírodovědné inženýrství

**Kombinovaný model proudění podzemní vody  
založený na primární formulaci MKP**

**Combined model of the groundwater flow based  
on primar formulation of FEM**

**Diplomová práce**

Autor:	Bc. Jan Palek
Vedoucí práce:	Ing. Otto Severýn, Ph.D.
Konzultant:	Ing. Jiřina Královcová, Ph.D.

**V Liberci 16. 5. 2008**

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum:

Podpis:

## **Poděkování**

Rád bych poděkoval vedoucímu mé diplomové práce panu Ing. Ottu Severýnovi, Ph.D. za jeho odbornou pomoc, cenné rady a poskytnuté informace.

Dále bych rád poděkoval panu Prof. Dr. Ing. Jiřímu Maryškovi, CSc. za jeho odbornou pomoc týkající se matematické části této práce a paní Ing. Jiřině Královcové, Ph.D. za poskytnuté síť na testování.

## **Anotace**

Tato diplomová práce se zabývá vývojem modelu proudění podzemní vody, který je realizován programem Flow123D a je založen na primární formulaci MKP. Bylo nutné navrhnout a implementovat algoritmus, který zajistí správnou funkčnost modelu pro multidimenzionální síť.

Po seznámení s použitým fyzikálním a matematickým modelem uvedeným v první kapitole se věnujeme popisu návrhu a implementace zmíněného modelu do programu Flow123D. Implementovaný algoritmus je v závěru práce otestován na benchmarkových úlohách.

Program Flow123D, testovací úlohy i software pro vykreslení výsledků modelu je možné nalézt na příložením CD.

**Klíčová slova:** primární formulace, MKP, proudění podzemní vody, multidimenzionální přístup

## **Abstract**

The aim of this Diploma Thesis is to develop a model of the groundwater flow. The model of this physical process is realized by program Flow123D and it is based on primar formulation of the FEM It was necessary to create and to implement an algorithm for correct calculation in case of multidimensional meshes. Used physical and mathematical model is briefly explained in the first chapter. The second chapter gives a concept and description of implementation of the new algorithm into the Flow123D program. There is tested the correct functionality of the implemented algorithm on several benchmark problems in the third chapter.

Flow123D program, input files, results of testing problems and visualization software is possible to find on enclosed CD.

**Keywords:** primar formulation, FEM, groundwater flow, multidimensional approach

# Obsah

<b>OBSAH .....</b>	<b>- 6 -</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>- 7 -</b>
<b>ÚVOD.....</b>	<b>- 8 -</b>
<b>1 MODEL PROUDĚNÍ PODZEMNÍCH VOD .....</b>	<b>- 11 -</b>
1.1 Fyzikální popis úlohy .....	- 11 -
1.2 Slabá formulace řešení, aplikace MKP .....	- 14 -
1.3 Počítačová implementace modelu .....	- 19 -
<b>2 NÁVRH A IMPLEMENTACE ALGORITMU .....</b>	<b>- 21 -</b>
2.1 Návrh algoritmu pro kombinaci 2D a 1D elementů .....	- 23 -
2.2 Návrh algoritmu pro kombinaci 3D a 2D elementů .....	- 26 -
<b>3 IMPLEMENTACE ALGORITMU .....</b>	<b>- 29 -</b>
3.1 Použití primárního modelu.....	- 29 -
3.2 Algoritmus z hlediska programátora.....	- 30 -
3.2.1 Multinode_test .....	- 32 -
3.2.2 Add_nodes .....	- 32 -
3.2.3 Create_elements.....	- 33 -
3.2.4 Assign_multinodes .....	- 33 -
3.2.5 Alocate_new_nodes.....	- 33 -
3.2.6 Fill_new_nodes.....	- 33 -
3.2.7 Area2D.....	- 34 -
3.2.8 Area3D.....	- 35 -
3.2.9 Create_msh_file .....	- 36 -
3.2.10 Create_bcd_file .....	- 36 -
3.2.11 Topology .....	- 36 -
3.2.12 Multi_n_row .....	- 37 -
3.2.13 Multi_fill_col_num .....	- 37 -
3.2.14 Multi_values .....	- 38 -
<b>4 TESTOVACÍ ÚLOHY .....</b>	<b>- 39 -</b>
4.1 Test z úvodu .....	- 39 -
4.2 Test 2D-1D na rozsáhlejší oblasti.....	- 40 -
4.3 Test 2D-3D na rozsáhlejší oblasti.....	- 42 -
<b>5 ZÁVĚR .....</b>	<b>- 51 -</b>
<b>LITERATURA.....</b>	<b>- 52 -</b>

# Seznam obrázků

Obr. 1-1, síť.....	- 9 -
Obr. 1-2, výsledek kombinace 1D a 2D.....	- 9 -
Obr. 1-3 - kompatibilní spojení 2D a 1D .....	- 20 -
Obr. 1-4 - nekompatibilní spojení 2D a 1D .....	- 20 -
Obr. 1-5 - kompatibilní spojení 3D a 2D .....	- 20 -
Obr. 1-6 - nekompatibilní spojení 3D a 2D .....	- 20 -
Obr. 2-1, sestavení globální matice .....	- 21 -
Obr. 2-2, společné uzly pro 1D a 2D elementy .....	- 22 -
Obr. 2-3, princip oddělení elementů 1D a 2D.....	- 23 -
Obr. 2-4, znázornění přidávání krajních uzlů 1D elementu.....	- 25 -
Obr. 2-5, požadovaná síť .....	- 25 -
Obr. 2-6, princip oddělení elementů 2D a 3D.....	- 26 -
Obr. 3-1, vývojový diagram.....	- 29 -
Obr. 3-2, 1. fáze, vývojový diagram .....	- 31 -
Obr. 3-3, Area2D .....	- 34 -
Obr. 3-4, Area3D .....	- 35 -
Obr. 3-5, 2.fáze, vývojový diagram .....	- 37 -
Obr. 4-2, síť.....	- 39 -
Obr. 4-3, síť.....	- 40 -
Obr. 4-4, $\bar{\sigma} = 0.01$ .....	- 41 -
Obr. 4-5, $\bar{\sigma} = 1.0$ .....	- 41 -
Obr. 4-6, $\bar{\sigma} = 10.0$ .....	- 42 -
Obr. 4-7, síť.....	- 43 -
Obr. 4-8, puklina v síti .....	- 43 -
Obr. 4-9, $\bar{\sigma} = 0.001$ ,Primar.....	- 44 -
Obr. 4-10, $\bar{\sigma} = 0.001$ ,MH .....	- 44 -
Obr. 4-11, $\bar{\sigma} = 1.0$ ,Primar.....	- 45 -
Obr. 4-12, $\bar{\sigma} = 1.0$ ,MH .....	- 45 -
Obr. 4-13, $\bar{\sigma} = 10.0$ ,Primar.....	- 46 -
Obr. 4-14, $\bar{\sigma} = 10.0$ ,MH .....	- 46 -
Obr. 4-15, $\bar{\sigma} = 0.001$ ,Primar.....	- 47 -
Obr. 4-16, $\bar{\sigma} = 0.001$ ,MH .....	- 47 -
Obr. 4-17, $\bar{\sigma} = 1.0$ ,Primar.....	- 48 -
Obr. 4-18, $\bar{\sigma} = 1.0$ ,MH .....	- 48 -
Obr. 4-19, $\bar{\sigma} = 10.0$ ,Primar.....	- 49 -
Obr. 4-20, $\bar{\sigma} = 10.0$ ,MH .....	- 49 -

# ÚVOD

Tématem této diplomové práce je vývoj modelu proudění podzemních vod. Model je založen na primární formulaci metody konečných prvků (MKP). Modelování tohoto přírodního jevu je základem pro řešení hydrogeologických, geochemických a hlavně ekologických problémů.

Pro tyto účely byl na katedře modelování procesů fakulty Mechatroniky a mezipředmětových inženýrských studií a dodnes stále je vyvíjen program Flow123D, více představen v dalším textu, nebo v [2]. Původně byl výpočet aproximovaného řešení ustáleného filtračního proudění založen pouze na smíšené hybridní (MH) formulaci MKP, více v [5]. Do Flow123D byly v rámci této formulace implementovány možnosti použití 1D elementů (linie), 2D elementů (trojúhelníky) a 3D elementů (čtyřstěny) i jejich kombinace, tzn. umí pracovat s oblastmi, ve kterých se pro 3D prostředí nacházejí kromě porézního média aproximovaného 3D elementy i pukliny aproximované 2D elementy a pro 2D prostředí je porézní médium aproximováno 2D elementem a puklina 1D elementem. Hovoříme o tzv. *multidimenzionálním přístupu a multidimenzionálních sítích*.

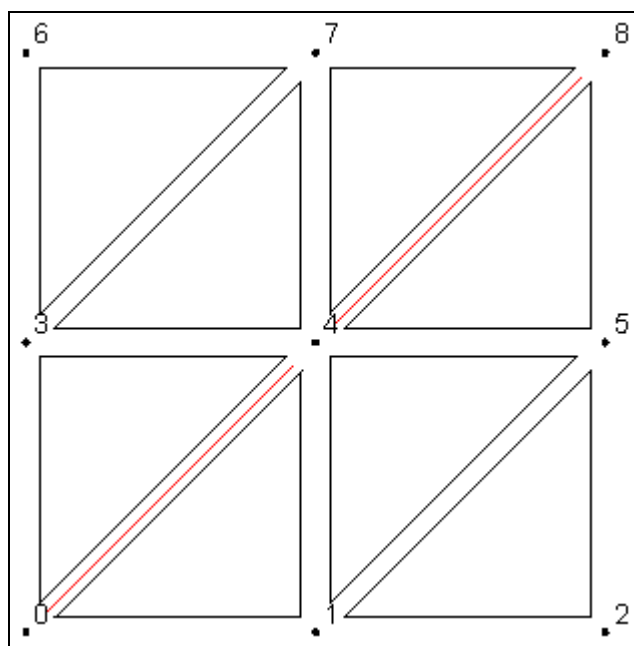
Později byla do programu z důvodů nevýhod (MH) formulace (mj. vede na velké indefinitní matice) implementována i primární formulace MKP, podrobněji v [6]. I v rámci této formulace byly do programu implementovány stejné tři prvky zastupující každou dimenzi. Funkčnost této implementace byla ověřena na několika základních testovacích úlohách. Testy byly prováděny nejprve na těch úlohách, jejichž síť obsahovaly pouze elementy jedné dimenze, výsledky těchto aproximací byly ve všech třech případech shodné s analytickými výsledky. Druhým typem testů byly testy na multidimenzionálních sítích. V některých situacích byly výsledky korektní, v jiných nikoliv. Konkrétně v případě příčného přetoku přes pukliny – ať v kombinaci 3D a 2D nebo v kombinaci 2D a 1D elementů – již docházelo k rozporům numerických výsledků s výsledky předpokládanými. Tok přes puklinu byl nezávislý na propustnosti pukliny, což neodpovídalo reálnému chování.

Pro názornost zde uvedeme elementární příklad, na kterém je zřetelně vidět výše zmíněný nedostatek.

Mějme síť na obr.1-1, Dirichletovy okrajové podmínky v uzlech  $P_6 = 10$  m,  $P_2 = 1$  m – těmi docílíme toku z levého horního rohu do pravého dolního rohu sítě,

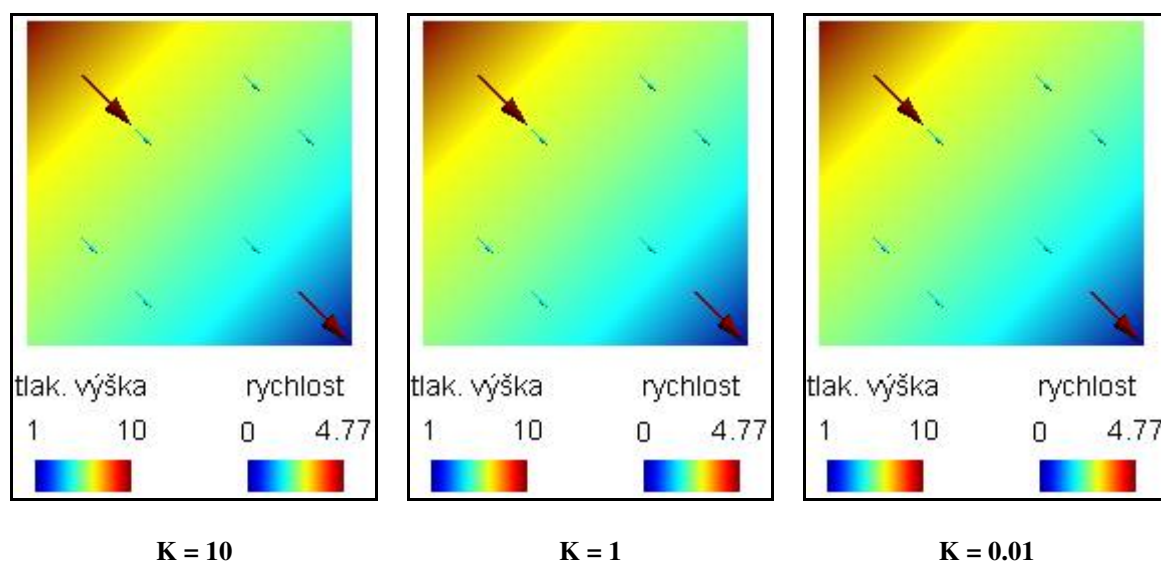
propustnost na 2D elementech  $K_T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  m/den, na 1D elementech (na obr. 1-1 jsou znázorněny červeně) postupně  $K_L = 10$  m/den,  $K_L = 1$  m/den a  $K_L = 0.01$  m/den.

**Sít'**(elementy jsou pro názornost zmenšeny a uzly posunuty):



Obr. 1-1, sít'

**Výpočet tlakových polí a vektorů rychlostí:**



Obr. 1-2, výsledek kombinace 1D a 2D



Z obr.1-2 je patrné, že výsledek je stále stejný, není zde žádná závislost na propustnosti pukliny. Intuitivně se dá očekávat, že málo propustná puklina bude v tomto případě fungovat jako hydraulický izolátor obou částí.

Tímto jsme se dostali k úkolům, které má tato práce vyřešit:

- Navrhnout algoritmus pro aproximaci úlohy filtračního proudění pomocí primární MKP a pro přidávání uzlů sítě na styku elementů různých dimenzí a pro stanovení přetoku mezi takovými uzly
- Implementovat navržené algoritmy do programu Flow123D
- Otestovat tyto algoritmy na modelových úlohách

Práce je rozčleněna do třech kapitol. V první kapitole je uveden fyzikální popis problému a jeho aproximace pomocí primární formulace MKP a seznámení s programem Flow123D, do kterého byla provedena implementace našeho algoritmu.

Ve druhé kapitole je nejprve popsán způsob, na kterém byla navržena komunikace elementů různých dimenzí v primární formulaci původně [6] a dále navržen postup, kterým je dosaženo korektního chování modelu. V této kapitole je následně popsán algoritmus jak z hlediska uživatele programu Flow123D, tak i z hlediska programátora algoritmu.

Ve třetí kapitole jsou uvedeny testovací úlohy, na kterých je ukázána funkčnost implementovaného algoritmu.

# 1 MODEL PROUDĚNÍ PODZEMNÍCH VOD

V této kapitole je zpracován základní matematicko-fyzikální popis filtračního proudění. Kapitola je rozdělena do tří částí:

- Fyzikální popis úlohy
- Slabá formulace řešení, aplikace MKP
- Počítačová implementace modelu

## 1.1 Fyzikální popis úlohy

Filtrační proudění kapaliny budeme studovat v oblasti  $\Omega$ , která je omezena hranicí  $\partial\Omega$ . Fyzikální popis úlohy je dán dvěma zákony:

### 1. Darcyho zákonem:

$$\mathbf{u} = -\mathbf{K}\nabla p \text{ na } \Omega, \quad (1)$$

kde  $\mathbf{u} [ms^{-1}]$ ... vektor filtrační rychlosti proudění,  $\mathbf{K} [ms^{-1}]$ ... tenzor propustnosti a  $\nabla p$  ... gradient tlakové výšky. Uvažujeme lineární Darcyho zákon ( $\mathbf{K}$  není funkcí  $p$ , ale pouze polohy).

### 2. rovnicí kontinuity:

$$\nabla \cdot \mathbf{u} = q \text{ na } \Omega, \quad (2)$$

kde  $q [s^{-1}]$ ... hustota zdrojů kapaliny

Rovnice kontinuity je vyjádřením zákona zachování hmoty respektive zákona zachování objemu pro nestlačitelnou tekutinu.

K těmto rovnicím je dále nutné uvažovat okrajové podmínky. V našem modelu využíváme dva druhy okrajových podmínek:

- Dirichletova okrajová podmínka:

$$p = p_D \text{ na } \partial\Omega_D, \quad (3)$$

která předepisuje hodnotu tlakové výšky na zadané části hranice.

- Neumannova okrajová podmínka:

$$\mathbf{u} \cdot \mathbf{n} = u_N \text{ na } \partial\Omega_N, \quad (4)$$

která předepisuje hodnotu filtračního toku na zbývajících částech hranice. Zde  $p_D [m]$ ...hodnoty tlakové výšky na hranici oblasti,  $u_N [ms^{-1}]$ ...hodnoty toku přes hranici oblasti,  $\mathbf{n}$ ... jednotkový vektor vnější normály křivky tvořící část hranice  $\partial\Omega_N$ .

Dirichletova okrajová podmínka modelované reálné oblasti je zadávána na liniích toků řek, na hladinách jezer, v měřených vrtech a podobně.

Neumannovu okrajovou podmínku obvykle zadáváme zejména pro vyjádření nepropustné hranice odpovídající nulovému toku (homogenní Neumannova okrajová podmínka).

Pro úplnost zde ještě zmíníme nejobecnější okrajovou podmínku, kterou však náš model nevyužívá. Jedná se o Newtonovu okrajovou podmínku, která vyjadřuje stav, kdy vnější tok hranicí závisí na spádu tlakové výšky a její tvar je:

$$\mathbf{u} \cdot \mathbf{n} = \sigma(p - p_D) \text{ na } \partial\Omega_W, \quad (5)$$

kde  $\sigma [s^{-1}]$  je přestupový koeficient.

Filtrační proudění kapaliny představuje průtok kapaliny oblastí  $\Omega$ , kde tato oblast je vyplněna porézním médiem – strukturou složenou ze zrn nebo vláken pevné látky, mezi nimiž je volný prostor (póry), který může být obsahovat vzduch nebo kapalinu. Dále může oblast  $\Omega$  obsahovat pukliny. Ty mohou být vyplněny málo propustným materiálem a tím tvořit hydraulický izolátor nebo naopak velmi dobře propouštět kapalinu a tím být hydraulickým kolektorem. V našem případě je vždy celý pórový i puklinový prostor zaplněn kapalinou a říkáme mu saturované prostředí. V této práci se zabýváme pouze stacionárním filtračním prouděním.

To, jak je porézní prostředí pro kapalinu propustné či nepropustné, vyjadřuje vlastnost – hydraulická vodivost značená  $\mathbf{K}$ . Obecně se jedná o pozitivně definitní tenzor druhého řádu. Pokud je prostředí homogenní a navíc izotropní, lze vlastnost daného prostředí vyjádřit jen jedním nezáporným reálným číslem  $\mathbf{a}$ . Pro propustnost pak platí:

$$\mathbf{K} = \mathbf{aE}, \quad (6)$$

kde  $\mathbf{E}$  je jednotková matice opět řádu shodného s dimenzí oblasti.

Cílem našeho modelu je získat hodnoty tlakových výšek  $p$  [m] v oblasti  $\Omega$ . Tlaková výška je výška volné hladiny nad referenčním místem. Pro upřesnění uvedeme, že tlaková výška vychází z obecnější piezometrické výšky  $\Phi$  [m], pro kterou byl Darcyho zákon původně odvozen. Její tvar je:

$$\Phi = h + \frac{p^*}{\rho g} \quad (7)$$

kde  $h$  [m] je referenční výška,  $p^*$  [Pa] je tlak,  $\rho$  [ $\text{kg} \cdot \text{m}^{-3}$ ] je hustota kapaliny a  $g$  [ $\text{m} \cdot \text{s}^{-2}$ ] je tíhové zrychlení. Výraz  $\frac{p^*}{\rho g}$  [m] vyjadřuje tlakovou výšku  $p$ . Darcyho zákon vyjádřen pomocí piezometrické výšky je:

$$\mathbf{u} = -\tilde{\mathbf{K}}\rho g \nabla \left( h + \frac{p^*}{\rho g} \right) \quad (8)$$

kde  $\tilde{\mathbf{K}}$  je tenzor propustnosti, hustota kapaliny je konstantní a stejně tak gravitační zrychlení považujeme v celé oblasti  $\Omega$  za konstantní. Pokud dále za referenční výšku  $h$  dosazujeme nulovou výšku, lze Darcyho zákon vyjádřit způsobem jako je ve vztahu (1), kde je tenzor propustnosti  $\mathbf{K}$  pouze korekcí pomocí hustoty a gravitačního zrychlení:

$$\mathbf{K} = \tilde{\mathbf{K}}\rho g \quad (9)$$

Z vypočtených tlakových výšek je dále třeba vypočítat i pole rychlosti filtračního proudění  $\mathbf{u}$   $[ms^{-1}]$  v naší oblasti  $\Omega$ . I když zde uvádíme jednotky rychlosti v  $[ms^{-1}]$ , ve skutečnosti se tato veličina pohybuje řádově v  $[cm \cdot den^{-1}]$ .

Výpočtem filtrační rychlosti a tlakové výšky získáme potřebné informace k posouzení hydrogeologické situace na oblasti  $\Omega$ .

## 1.2 Slabá formulace řešení, aplikace

### MKP

V tomto odstavci si ukážeme, jak v případě naší konkrétní úlohy filtračního proudění dospějeme od matematicko fyzikálního modelu uvedeného v předchozí kapitole k modelu numerickému. Cílem této práce není přesné matematické odvození modelu, proto jsou uvedeny pouze hlavní kroky tohoto odvození.

Nejprve definujeme v dalším postupu použité prostory funkcí:

$$L_2(\Omega) \quad (10)$$

je prostor funkcí integrovatelných s druhou mocninou na oblasti  $\Omega$ .

$$H^1(\Omega) = \left\{ w \in L_2(\Omega); \int_{\Omega} (|w|^2 + |\nabla w|^2) dx < \infty \right\} \quad (11)$$

je Sobolevův prostor funkcí  $w \in L_2(\Omega)$  kvadraticky integrabilní včetně svého gradientu. Pro formulaci slabého řešení potřebujeme dále prostor:

$$H_0^1(\Omega) = \{ w \in H^1(\Omega); \mathcal{W}w = 0 \in \partial\Omega \} \quad (12)$$

kde  $\mathcal{V}$  je operátor stop vytvářející restrikcí na hranici oblasti. Dále pro tu část hranice oblasti  $\partial\Omega_D$ , kde je předepsána Dirichletova okrajová podmínka, definujeme prostor:

$$H_{D,*}^1(\Omega) = \{ w \in H^1(\Omega); \mathcal{W}w = p_D \in \partial\Omega_D \} \quad (13)$$

v tomto prostoru jsou splněny Dirichletovy okrajové podmínky, zatímco:

$$H_{D,0}^1(\Omega) = \{w \in H^1(\Omega); \gamma w = 0 \in \partial\Omega_D\} \quad (14)$$

je prostor funkcí, které mají na hranici nulové stopy.

Dále si pro zjednodušení zápisu zavádíme skalární součin funkcí v prostoru  $L_2(\Omega)$ :

$$(u, v) = \int_{\Omega} u(x)v(x)dx \quad (15)$$

skalární součin funkcí v prostoru  $L_2(\partial\Omega)$ :

$$\langle u, v \rangle = \int_{\partial\Omega} u(x)v(x)dx \quad (16)$$

Nyní se budeme věnovat již samotnému postupu odvození modelu.

Pokud Darcyho zákon dosadíme do rovnice kontinuity, získáme tím rovnici v divergentním tvaru:

$$-\nabla \cdot (K\nabla p) = q \quad (17)$$

Obě strany rovnice vynásobíme testovací funkcí  $w \in H_0^1(\Omega)$  a integrujeme přes oblast  $\Omega$ , získáme rovnici ve tvaru:

$$(-\nabla \cdot (K\nabla p), w)_{\Omega} = (q, w)_{\Omega} \quad (18)$$

Na vztah (18) aplikujeme Greenovu větu a získáme:

$$\langle -K\nabla p \cdot n, w \rangle + (K\nabla p, \nabla w)_{\Omega} = (q, w)_{\Omega} . \quad (19)$$

Kde člen  $K\nabla p \cdot n$  je zadaný tok  $u_N$  na hranici oblasti, který převedeme na pravou stranu:

$$(K\nabla p, \nabla w)_{\Omega} = (q, w)_{\Omega} + \langle u_N, w \rangle_{\partial\Omega_N} . \quad (20)$$

**Definice:**

Slabým řešením nazveme takovou funkci  $p \in H_{D,*}^1(\Omega)$ , která splňuje rovnici (20) pro všechny testovací funkce  $w \in H_{D,0}^1(\Omega)$ .

Nyní z důvodu řešitelnosti úlohy převedeme nekonečně dimenzionální problém na konečně dimenzionální použitím numerické metody – metody konečných prvků. Aproximace řešení spočívá v rozdělení spojitě oblasti na konečný počet prvků (elementů), tyto elementy jsou jednoznačně určeny uzly (vrcholy elementů). V primární formulaci hledáme hodnoty tlakových výšek na uzlech elementů.

Z matematického hlediska spočívá aproximace řešení rovnice (20) v nahrazení hledané funkce  $p$  lineární kombinací báзовých funkcí. Ze stejného důvodu je vhodné mít zadány v lineární kombinaci báзовých funkcí i funkce zdrojů  $q$  a funkce zadaných toků  $u$  :

$$p = \sum_{i \in I} P^i \cdot \varphi_i, \quad q = \sum_{i \in I} Q^i \cdot \varphi_i \quad \text{a} \quad u = \sum_{i \in I_N} G^i \cdot \varphi_i \quad (21)$$

kde  $\varphi_i \in H_h^1(\Omega)$  je báзовá funkce a  $H_h^1(\Omega)$  je podprostor  $H^1(\Omega)$

s diskretizačním parametrem  $h$  a  $P^i, Q^i$  a  $G^i$  jsou hodnoty tlakových výšek, zdrojů a toků, přičemž hodnoty tlakových výšek, které nejsou zadány okrajovou podmínkou, je nutné dopočítat, ostatní hodnoty jsou z okrajových podmínek známe nebo v našem případě hodnoty  $Q^i$  považovány za nulové (jako nulové zdroje v oblasti), ty je tedy možné z výpočtu vynechat.

Po nahrazení testovací funkce báзовou  $w = \varphi$  dosadíme vztahy (21) do rovnice (20) a dostaneme:

$$\sum_{i \in I - I_D} P^i (K \nabla \varphi_i, \nabla \varphi_j) + \sum_{i \in I_D} P^i (K \nabla \varphi_i, \nabla \varphi_j) = \sum_{i \in I} Q^i (\varphi_i, \varphi_j) + \sum_{i \in I_N} G^i (\varphi_i, \varphi_j) \quad (22)$$

kde  $I$  je množina všech indexů odpovídající uzlům elementů diskretizované oblasti  $\Omega$ .  $I_N$  je množina indexů na hranici oblasti, kde je zadána Neumannova okrajová

podmínka a  $I_D$  je množina indexů na hranici oblasti, kde je zadána Dirichletova okrajová podmínka.

Dále lze členy  $(\varphi_i, \varphi_j)$  nahradit maticí  $\Phi$ , koeficienty  $Q^i$  vektorem  $\vec{Q}$ , koeficienty  $G^i$  vektorem  $\vec{G}$ . Získáme tvar:

$$\sum_{i \in I-I_D} P^i (K \nabla \varphi_i, \nabla \varphi_j) + \sum_{i \in I_D} P^i (K \nabla \varphi_i, \nabla \varphi_j) = Q^T \Phi + G^T \Phi \quad (23)$$

A součet členů na pravé straně nahradíme jedním vektorem  $b$ .

$$Q^T \Phi + G^T \Phi = b \quad (24)$$

Dále se budeme zabývat levou stranou rovnice (22). V ní vystupují dva členy, člen

$$\sum_{i \in I-I_D} P^i (K \nabla \varphi_i, \nabla \varphi_j)$$

obsahuje neznámé hodnoty  $P^i$  a člen

$$\sum_{i \in I_D} P^i (K \nabla \varphi_i, \nabla \varphi_j)$$

obsahuje známé hodnoty tlakových výšek, zadané v Dirichletově okrajové podmínce. Prozatím člen se zadanými okrajovými podmínkami vynecháme, člen rovnice (22)  $(K \nabla \varphi_i, \nabla \varphi_j)$  nahradíme maticí  $\tilde{A}$ , známé i neznámé koeficienty  $P^i$  lze vyjádřit jako vektor  $P$ . Získáme tak systém stavových rovnic:

$$\tilde{A} \cdot P = \tilde{b} \quad (25)$$

Do systému (11) ještě chybí dodat Dirichletovy okrajové podmínky. Pro případ zadané tlakové výšky v uzlu je potřeba vzít všechny nenulové prvky matice  $\tilde{A}$  daného uzlu (každý uzel má jeden řádek a jeden sloupec v matici) a jednotlivě je vynásobit se zadanou hodnotou tlakové výšky a odečíst od příslušného řádku vektoru pravé strany. Následně je nutné vynulovat všechny prvky v řádku a ve sloupci náležící danému uzlu a pouze na diagonálu vložit hodnotu 1. Na řádek vektoru pravé strany, který patří



danému uzlu, se vloží zadaná hodnota tlakové výšky. Po provedení tohoto algoritmu získáme konečný tvar lineárního systému rovnic:

$$A \cdot P = b \quad (26)$$

Vektor neznámých hodnot tlakových výšek  $P$  vypočítán řešičem lineárních systémů, v našem případě je využit Matlab.

Matici  $A$  nazýváme stavovou nebo také globální maticí, je složena z tzv. lokálních matic, což jsou jednotlivé příspěvky k výrazu  $(K \nabla \varphi_i, \nabla \varphi_j)$  počítané pro jednotlivé elementy celé diskretizované oblasti (sítě).

Bázové funkce používáme lineární ve tvaru závislém na dimenzi elementu. Pro úsečku:

$$\varphi_i(x) = \alpha_i + \beta_i x \quad (27)$$

pro trojúhelník:

$$\varphi_i(x, y) = \alpha_i + \beta_i x + \gamma_i y \quad (28)$$

a pro čtyřstěn:

$$\varphi_i(x, y, z) = \alpha_i + \beta_i x + \gamma_i y + \delta_i z \quad (29)$$

Její koeficienty  $\alpha$ ,  $\beta$ ,  $\gamma$  a případně  $\delta$  dopočítáváme ze známých hodnot, které musí básová funkce splňovat nad danými uzly jednotlivých elementů. Funkce musí splňovat tuto podmínku:

$$\varphi_i(x_j) = \delta_{ij} \quad (30)$$

kde  $\varphi_i$  je básová funkce  $i$ -tého uzlu,  $x_j$  je  $j$ -tý uzel,  $\delta_{ij}$  je Kroneckerovo delta ( $= 1$  pro  $i = j$ ;  $= 0$  pro  $i \neq j$ ).

Detailnější rozepsání tvarů básových funkcí, výpočtů jejich koeficientů  $i$  struktury lokálních matic jednotlivých elementů je možné najít v [6].

Všeobecně lze uvést, že výhodou primární formulace je její relativně snadná realizace – jednoduché odvození formulace úlohy a nepříliš složitý algoritmus sestavení globální matice. Diskretizací rovnice řešíme soustavu pro primární veličinu  $p$  s menším

počtem proměnných, než tomu je ve smíšené nebo smíšené-hybridní formulaci MKP a navíc je stavová matice primární formulace symetricky pozitivně definitní a tím urychluje možnost výpočtu na rozdíl od MH formulace, jejíž stavová matice je indefinitní.

Nevýhodou však je skutečnost, že vypočteme primární veličinu  $p$  a filtrační rychlost  $\mathbf{u}$  musíme následně vypočítat zpětným dosazením do Darcyho zákona. Tím je snížena hladkost, filtrační rychlost se stává po elementech konstantní. Další nevýhodou je, že bilanční rovnice je v této formulaci splněna v obecném případě na celé oblasti a v každé oblasti, která obklopuje uzel, ale není splněna na jednotlivých elementech v síti, a proto není možné pomocí této formulace počítat transport látky, který tento předpoklad vyžaduje.

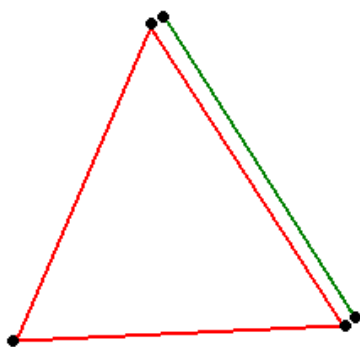
Přesto poskytuje dostatečné výsledky k posouzení hydrogeologické situace.

## 1.3 Počítačová implementace modelu

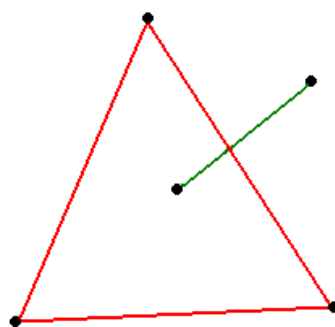
Po seznámení s matematicko fyzikálním modelem uvedeném v předchozí kapitole si představíme program Flow123D, který je počítačovou implementací tohoto modelu. Je psán v programovacím jazyku C a v současné době umožňuje kromě výpočtu ustáleného proudění také simulaci neustáleného proudění nebo transportu látek v podzemí. V naší práci vyvíjíme tu část programu, která se zabývá výpočtem ustáleného proudění.

Pro náš model je navíc tento program výhodný z toho důvodu, že je v něm již implementován algoritmus pro vytvoření potřebného \*.ngh souboru, seznamu všech sousedností elementů z celé sítě. Postup spuštění tohoto algoritmu, stejně tak postup spuštění výpočtu ustáleného proudění je uveden v příloze této práce.

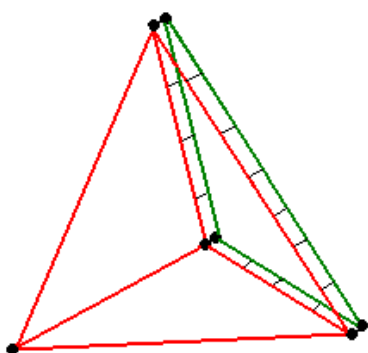
Charakteristickou vlastností programu Flow123D je použití kombinovaného (multidimenzionálního) přístupu. Používá prvky tří dimenzí, kde 3D odpovídá poréznímu médiu, 2D puklinám a 1D kanálům. Program umožňuje dva způsoby propojení a to buď kompatibilní, nebo nekompatibilní. Prvky různých dimenzí jsou propojeny kompatibilně tehdy, má-li prvek s vyšší dimenzí celou jednu stěnu společnou právě s celým jedním prvkem nižší dimenze. Pro ilustraci jsou uvedeny příklady spojení. V rozsáhlých úlohách je mnohdy nevyhnutelné prvky spojovat nekompatibilně a to z důvodu vysoké algoritmické náročnosti kompatibilního spojení prvků. V této práci je uvažováno pouze kompatibilní spojování.



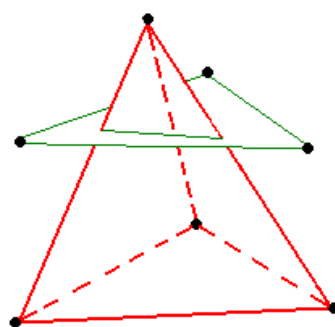
Obr. 1-3 - kompatibilní spojení 2D a 1D



Obr. 1-4 - nekompatibilní spojení 2D a 1D



Obr. 1-5 - kompatibilní spojení 3D a 2D



Obr. 1-6 - nekompatibilní spojení 3D a 2D

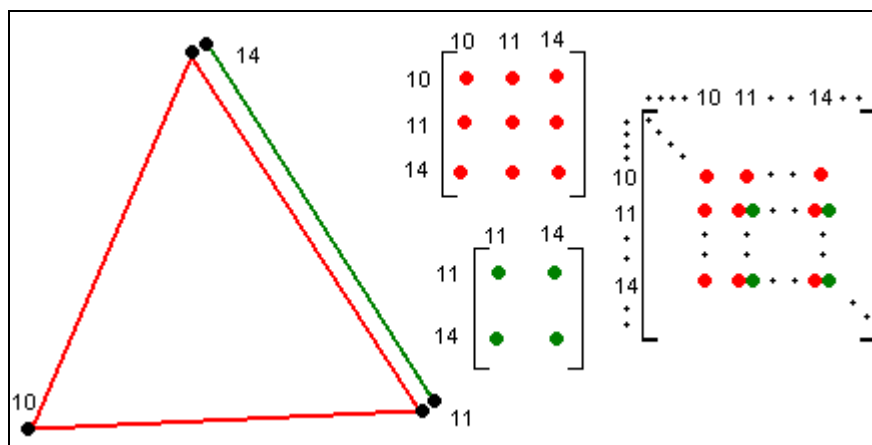
## 2 NÁVRH A IMPLEMENTACE ALGORITMU

V této části práce si rozdělíme problém na dvě části:

- Návrh algoritmu pro kombinaci 2D a 1D elementů
- Návrh algoritmu pro kombinaci 3D a 2D elementů

Před popisem zmíněných částí této práce je však třeba vysvětlit způsob, jakým byl realizován kombinovaný model doposavad.

Původní implementace primární formulace má komunikaci mezi elementy různých dimenzí uskutečněnou díky společným uzlům sousedících elementů. Do globální matice se zapisují prvky lokálních matic všech elementů a to i různých dimenzí, jak je naznačeno na obr. 2-1. Prvky lokální matice 1D elementu (zeleně) jsou v globální matici sečteny s příslušnými prvky lokální matice 2D elementu (červeně).

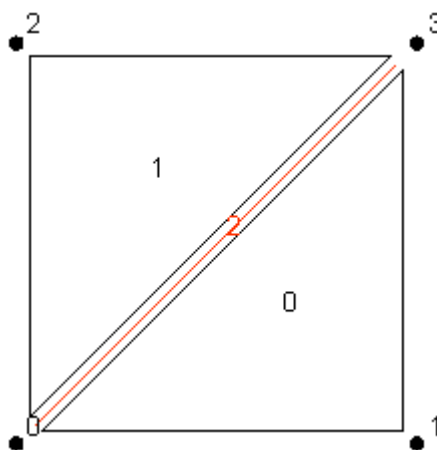


Obr. 2-1, sestavení globální matice

Způsob, jakým je dosaženo komunikace mezi 2D a 3D elementem, je analogický k výše uvedenému, rozdíl je pouze v rozměru lokálních matic elementů, pro čtyřstěn je to matice čtvrtého řádu a pro trojúhelník zůstává matice třetího řádu, jako tomu je na obr. 2-1. Tím v globální matici sčítáme 9 pozic (příspěvek 2D elementu) namísto výše uvedených 4 pozic (příspěvek 1D elementu).

Podle výsledků testu pro příčný přetok přes element nižší dimenze (puklinu) z úvodu této práce je tento způsob komunikace nevhodný. Příčina spočívá v tomto: na obr. 2-2 jsou pro elementy 0, 1 a 2 společné uzly 0 a 3 a dochází v toku sítí napříč

liniovým elementem (např. od uzlu 2 k uzlu 1) a jelikož je primární formulace založena na výpočtu tlakových výšek na jednotlivých uzlech v síti, tak vypočtená tlaková výška na uzlu 0 a 3 je stejná jak pro liniový element, tak i pro oba trojúhelníkové elementy, nezávisle na propustnosti linie. Jednoduše vysvětleno, tok z elementu 1 „přeskočí“ rovnou na element 0 a nezohlední se tím již zmíněná propustnost elementu 2.

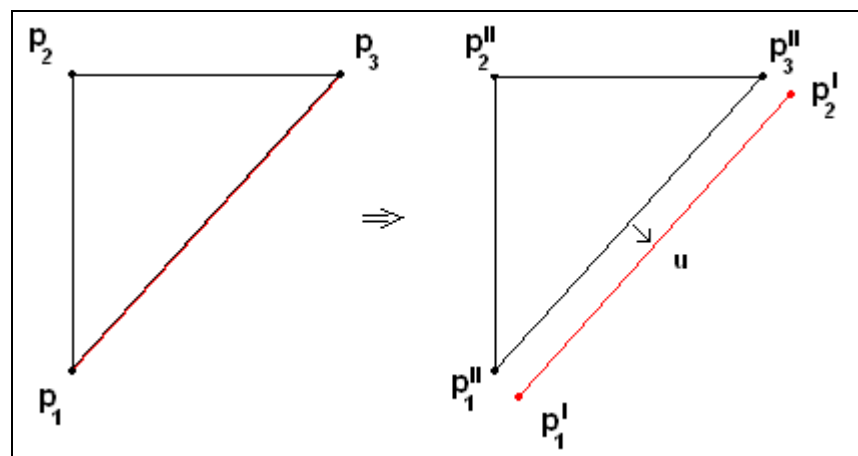


Obr. 2-2, společné uzly pro 1D a 2D elementy

Řešením tohoto problému je následující postup. Na obr. 2-3 je v levé části naznačeno původní spojení 2D a 1D elementů. Na pravé straně naznačujeme, že 1D element od 2D elementu fyzicky oddělíme. Tím je pro řádný výpočet třeba dodefinovat velikost přetoku mezi těmito elementy, který je úměrný rozdílu tlakových výšek. Použijeme tento vztah:

$$\mathbf{u} = \sigma_c \left( \frac{p_1'' + p_2'' + p_3''}{3} - \frac{p_1' + p_2'}{2} \right) \quad (31)$$

Kde  $\mathbf{u} [ms^{-1}]$  je přetok mezi 2D a 1D elementem,  $\sigma_c [s^{-1}]$  je přestupový koeficient mezi elementy,  $p_1'' [m]$  je tlaková výška v uzlu číslo 1 ve 2D elementu a  $p_1' [m]$  je tlaková výška v uzlu 1 v 1D elementu. První člen v závorce ve vztahu (31) je celková tlaková výška 2D elementu a druhý člen je celková tlaková výška 1D elementu. V podstatě se jedná o analogii Newtonovy okrajové podmínky uvedené ve vztahu (5) první kapitoly s tím rozdílem, že ji používáme pro přetok mezi elementy různých dimenzí uvnitř sítě.



Obr. 2-3, princip oddělení elementů 1D a 2D

Čím menší hodnota přestupového koeficientu  $\sigma_c$ , tím dochází u dané sousednosti elementů různých dimenzí k menšímu přetoku.

Tímto jsme uvedli způsob zpracování nové komunikace mezi elementy různých dimenzí obecně a v následujícím textu toto rozvedeme pro kombinaci 1D s 2D elementy a 2D s 3D elementy.

## 2.1 Návrh algoritmu pro kombinaci 2D a 1D elementů

Řešení spočívá ve vytvoření nových uzlů v místech napojení 2D elementů na 1D elementy, tím se docílí jednoznačného oddělení 1D a 2D elementů. Elementy spolu v modelu nebudou již propojeny přes společné uzly, ale přes přestupový koeficient  $\sigma_c$  charakterizující množství vody přeteklé mezi elementy (uzly).

Připomeňme, že primární model MKP vede na soustavu  $n$  rovnic o  $n$  neznámých v maticovém zápisu:

$$A \cdot P = b$$

Kde  $A$  je globální matice,  $P$  jsou hledané tlakové výšky v uzlech sítě a  $b$  je vektor pravé strany. Takovéto propojení elementů funguje jako zdroj  $q$ , který je na pravé straně rovnice.

Ze vztahu (31) podle obr.2-3 získáme na pravé straně rovnice (26):

$$\mathbf{u}_1^{\text{II}} = \mathbf{u}_2^{\text{II}} = \mathbf{u}_3^{\text{II}} = -\frac{1}{3}\sigma_c \left( \frac{p_1^{\text{II}} + p_2^{\text{II}} + p_3^{\text{II}}}{3} - \frac{p_1^{\text{I}} + p_2^{\text{I}}}{2} \right) \quad (32)$$

$$\mathbf{u}_1^{\text{I}} = \mathbf{u}_2^{\text{I}} = +\frac{1}{2}\sigma_c \left( \frac{p_1^{\text{II}} + p_2^{\text{II}} + p_3^{\text{II}}}{3} - \frac{p_1^{\text{I}} + p_2^{\text{I}}}{2} \right) \quad (33)$$

po rozepsání:

$$\mathbf{u}_1^{\text{II}} = -\frac{\sigma_c p_1^{\text{II}}}{9} - \frac{\sigma_c p_2^{\text{II}}}{9} - \frac{\sigma_c p_3^{\text{II}}}{9} + \frac{\sigma_c p_1^{\text{I}}}{6} + \frac{\sigma_c p_2^{\text{I}}}{6} \quad (34)$$

$$\mathbf{u}_2^{\text{II}} = -\frac{\sigma_c p_1^{\text{II}}}{9} - \frac{\sigma_c p_2^{\text{II}}}{9} - \frac{\sigma_c p_3^{\text{II}}}{9} + \frac{\sigma_c p_1^{\text{I}}}{6} + \frac{\sigma_c p_2^{\text{I}}}{6}$$

$$\mathbf{u}_3^{\text{II}} = -\frac{\sigma_c p_1^{\text{II}}}{9} - \frac{\sigma_c p_2^{\text{II}}}{9} - \frac{\sigma_c p_3^{\text{II}}}{9} + \frac{\sigma_c p_1^{\text{I}}}{6} + \frac{\sigma_c p_2^{\text{I}}}{6}$$

$$\mathbf{u}_1^{\text{I}} = +\frac{\sigma_c p_1^{\text{II}}}{6} + \frac{\sigma_c p_2^{\text{II}}}{6} + \frac{\sigma_c p_3^{\text{II}}}{6} - \frac{\sigma_c p_1^{\text{I}}}{4} - \frac{\sigma_c p_2^{\text{I}}}{4}$$

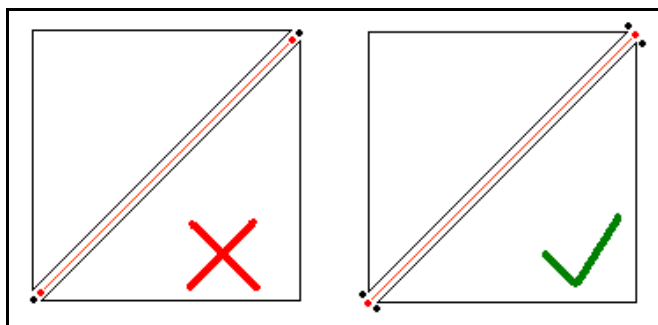
$$\mathbf{u}_2^{\text{I}} = +\frac{\sigma_c p_1^{\text{II}}}{6} + \frac{\sigma_c p_2^{\text{II}}}{6} + \frac{\sigma_c p_3^{\text{II}}}{6} - \frac{\sigma_c p_1^{\text{I}}}{4} - \frac{\sigma_c p_2^{\text{I}}}{4}$$

Výše uvedené vztahy převedeme na levou stranu rovnice (26) a v globální matici se projeví takto:

$$\left( \begin{array}{ccc|ccc|ccc} \hline \hline +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & & & & -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & & & \\ +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & & & & -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & & & \\ +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & & & & -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & & & \\ \hline & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ \hline -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & & & & +\frac{\sigma_c}{4} & +\frac{\sigma_c}{4} & & & \\ -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & & & & +\frac{\sigma_c}{4} & +\frac{\sigma_c}{4} & & & \\ -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & -\frac{\sigma_c}{6} & & & & +\frac{\sigma_c}{4} & +\frac{\sigma_c}{4} & & & \\ \hline \hline \end{array} \right) \begin{pmatrix} \overline{p_1^{\text{II}}} \\ \overline{p_2^{\text{II}}} \\ \overline{p_3^{\text{II}}} \\ \dots \\ \overline{p_1^{\text{I}}} \\ \overline{p_2^{\text{I}}} \\ \overline{p_3^{\text{I}}} \\ \dots \end{pmatrix} = \begin{pmatrix} \overline{r_1^{\text{II}}} \\ \overline{r_2^{\text{II}}} \\ \overline{r_3^{\text{II}}} \\ \dots \\ \overline{r_1^{\text{I}}} \\ \overline{r_2^{\text{I}}} \\ \overline{r_3^{\text{I}}} \\ \dots \end{pmatrix} \quad (35)$$

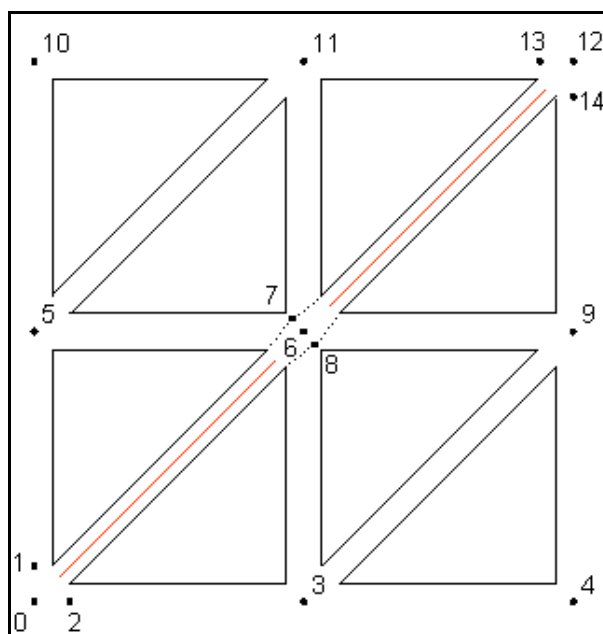
Všechny hodnoty v globální matici budou tedy přičteny (odečteny) na dané pozice ke stávajícím hodnotám získaným z lokálních matic elementů. Hodnoty  $r$  jsou hodnoty pravé strany příslušného uzlu, které vzniknou součtem původní pravé strany a

právě příslušným přetokem  $u$ . Počet uzlů, které je třeba kolem originálního uzlu vytvořit pro jednoznačné oddělení elementů různých dimenzí, je roven počtu liniových elementů připojených k danému uzlu. Výjimkou jsou krajní uzly 1D elementů, kde místo jednoho nového uzlu přidáváme uzly dva obr. 2-4. Nový uzel znamená uzel s identicky stejnými geometrickými vlastnostmi jako má originální uzel, tj. na obrázcích teoreticky nejsou nové uzly vidět, proto je pro přehlednost zobrazujeme s malým posunutím směrem k elementům, se kterými mají být spojeny.



Obr. 2-4, znázornění přidávání krajních uzlů 1D elementu

Pro názornost použijme originální síť z typového příkladu v úvodu. Vidíme, že uzly liniových elementů jsou zároveň uzly trojúhelníkových elementů. Cílem nového algoritmu je zobrazení na obr.2-5. Jak je z obrázku patrné, originální uzly ponecháváme liniovým elementům a nové uzly přiřazujeme trojúhelníkovým elementům. Každý uzel má své identifikační číslo, očíslování nových uzlů je provedeno tak, že se přechíslovuje celá síť od začátku i s násobnými uzly jako na obr. 2-5.



Obr. 2-5, požadovaná síť

Více je celý postup běhu algoritmu pospán níže, v kapitole 2.3.

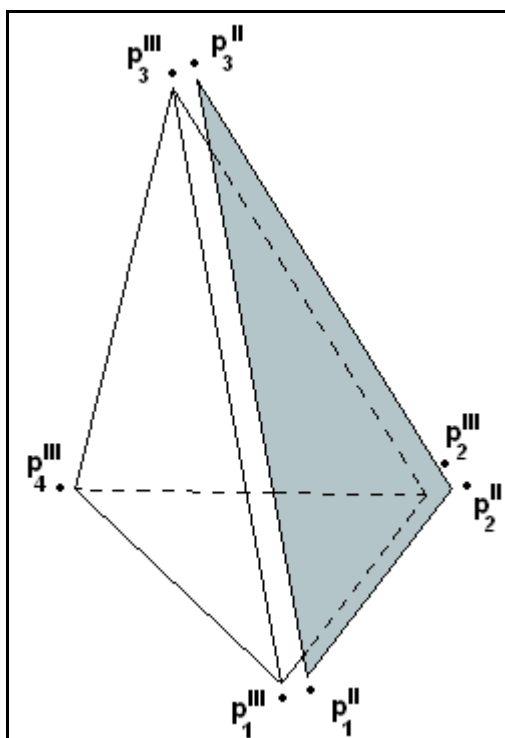


## 2.2 Návrh algoritmu pro kombinaci 3D a 2D elementů

U kombinace 2D a 3D elementů je situace analogická výše popsané kombinaci 1D a 2D elementů. Opět dostane každý element v místech styku s elementem jiné dimenze nový uzel. Rovnice (31) se přemění na:

$$\mathbf{u} = \sigma_c \left( \frac{p_1^{III} + p_2^{III} + p_3^{III} + p_4^{III}}{4} - \frac{p_1^{II} + p_2^{II} + p_3^{II}}{3} \right) \quad (36)$$

Kde  $\mathbf{u} [ms^{-1}]$  je přetok mezi 3D a 2D elementem,  $\sigma_c [s^{-1}]$  je přestupový koeficient mezi elementy,  $p_1^{III} [m]$  je tlaková výška v uzlu číslo 1 ve 3D elementu a  $p_1^{II} [m]$  je tlaková výška v uzlu 1 v 2D elementu. První člen v závorce ve vztahu (36) je celková tlaková výška 3D elementu a druhý člen je celková tlaková výška 2D elementu. Kombinace těchto dvou dimenzí je naznačena na obr.2-5.



Obr. 2-6, princip oddělení elementů 2D a 3D

Stejně jako u předešlé kombinace si odvodíme koeficienty, které se budou přičítat (odečítat) od stávajících hodnot na daných pozicích v globální matici.

Vztah (36) si opět přepíšeme do tvaru:

$$\mathbf{u}_1^{\text{III}} = \mathbf{u}_2^{\text{III}} = \mathbf{u}_3^{\text{III}} = \mathbf{u}_4^{\text{III}} = -\frac{1}{4} \sigma_c \left( \frac{p_1^{\text{III}} + p_2^{\text{III}} + p_3^{\text{III}} + p_4^{\text{III}}}{4} - \frac{p_1^{\text{II}} + p_2^{\text{II}} + p_3^{\text{II}}}{3} \right) \quad (37)$$

$$\mathbf{u}_1^{\text{II}} = \mathbf{u}_2^{\text{II}} = \mathbf{u}_3^{\text{II}} = +\frac{1}{3} \sigma_c \left( \frac{p_1^{\text{II}} + p_2^{\text{II}} + p_3^{\text{II}}}{3} - \frac{p_1^{\text{III}} + p_2^{\text{III}} + p_3^{\text{III}} + p_4^{\text{III}}}{4} \right) \quad (38)$$

Po rozepsání:

$$\mathbf{u}_1^{\text{III}} = -\frac{\sigma_c p_1^{\text{III}}}{16} - \frac{\sigma_c p_2^{\text{III}}}{16} - \frac{\sigma_c p_3^{\text{III}}}{16} - \frac{\sigma_c p_4^{\text{III}}}{16} + \frac{\sigma_c p_1^{\text{II}}}{12} + \frac{\sigma_c p_2^{\text{II}}}{12} + \frac{\sigma_c p_3^{\text{II}}}{12} \quad (39)$$

$$\mathbf{u}_2^{\text{III}} = -\frac{\sigma_c p_1^{\text{III}}}{16} - \frac{\sigma_c p_2^{\text{III}}}{16} - \frac{\sigma_c p_3^{\text{III}}}{16} - \frac{\sigma_c p_4^{\text{III}}}{16} + \frac{\sigma_c p_1^{\text{II}}}{12} + \frac{\sigma_c p_2^{\text{II}}}{12} + \frac{\sigma_c p_3^{\text{II}}}{12}$$

$$\mathbf{u}_3^{\text{III}} = -\frac{\sigma_c p_1^{\text{III}}}{16} - \frac{\sigma_c p_2^{\text{III}}}{16} - \frac{\sigma_c p_3^{\text{III}}}{16} - \frac{\sigma_c p_4^{\text{III}}}{16} + \frac{\sigma_c p_1^{\text{II}}}{12} + \frac{\sigma_c p_2^{\text{II}}}{12} + \frac{\sigma_c p_3^{\text{II}}}{12}$$

$$\mathbf{u}_4^{\text{III}} = -\frac{\sigma_c p_1^{\text{III}}}{16} - \frac{\sigma_c p_2^{\text{III}}}{16} - \frac{\sigma_c p_3^{\text{III}}}{16} - \frac{\sigma_c p_4^{\text{III}}}{16} + \frac{\sigma_c p_1^{\text{II}}}{12} + \frac{\sigma_c p_2^{\text{II}}}{12} + \frac{\sigma_c p_3^{\text{II}}}{12}$$

$$\mathbf{u}_1^{\text{II}} = +\frac{\sigma_c p_1^{\text{III}}}{12} + \frac{\sigma_c p_2^{\text{III}}}{12} + \frac{\sigma_c p_3^{\text{III}}}{12} + \frac{\sigma_c p_4^{\text{III}}}{12} - \frac{\sigma_c p_1^{\text{II}}}{9} - \frac{\sigma_c p_2^{\text{II}}}{9} - \frac{\sigma_c p_3^{\text{II}}}{9} \quad (40)$$

$$\mathbf{u}_2^{\text{II}} = +\frac{\sigma_c p_1^{\text{III}}}{12} + \frac{\sigma_c p_2^{\text{III}}}{12} + \frac{\sigma_c p_3^{\text{III}}}{12} + \frac{\sigma_c p_4^{\text{III}}}{12} - \frac{\sigma_c p_1^{\text{II}}}{9} - \frac{\sigma_c p_2^{\text{II}}}{9} - \frac{\sigma_c p_3^{\text{II}}}{9}$$

$$\mathbf{u}_3^{\text{II}} = +\frac{\sigma_c p_1^{\text{III}}}{12} + \frac{\sigma_c p_2^{\text{III}}}{12} + \frac{\sigma_c p_3^{\text{III}}}{12} + \frac{\sigma_c p_4^{\text{III}}}{12} - \frac{\sigma_c p_1^{\text{II}}}{9} - \frac{\sigma_c p_2^{\text{II}}}{9} - \frac{\sigma_c p_3^{\text{II}}}{9}$$

A protože tlakové výšky  $p$  jsou neznámé, výše uvedené vztahy převedeme na levou stranu rovnice (26) a v globální matici se projeví takto:

$$\left( \begin{array}{c|c|c} \begin{array}{cccc} +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} \\ +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} \\ +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} \\ +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} & +\frac{\sigma_c}{16} \end{array} & \begin{array}{ccc} -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \\ -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \\ -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \\ -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \end{array} & \begin{array}{c} \overline{\overline{\overline{p_1^{III}}}} \\ \overline{\overline{\overline{p_2^{III}}}} \\ \overline{\overline{\overline{p_3^{III}}}} \\ \overline{\overline{\overline{p_4^{III}}}} \end{array} \\ \hline \begin{array}{cccc} -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \\ -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \\ -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \\ -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} & -\frac{\sigma_c}{12} \end{array} & \begin{array}{ccc} +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} \\ +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} \\ +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} \\ +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} & +\frac{\sigma_c}{9} \end{array} & \begin{array}{c} \overline{\overline{\overline{p_1^{II}}}} \\ \overline{\overline{\overline{p_2^{II}}}} \\ \overline{\overline{\overline{p_3^{II}}}} \end{array} \end{array} \right) = \left( \begin{array}{c} \overline{\overline{\overline{r_1^{III}}}} \\ \overline{\overline{\overline{r_2^{III}}}} \\ \overline{\overline{\overline{r_3^{III}}}} \\ \overline{\overline{\overline{r_4^{III}}}} \\ \dots \\ \overline{\overline{\overline{r_1^{II}}}} \\ \overline{\overline{\overline{r_2^{II}}}} \\ \overline{\overline{\overline{r_3^{II}}}} \end{array} \right) \quad (41)$$

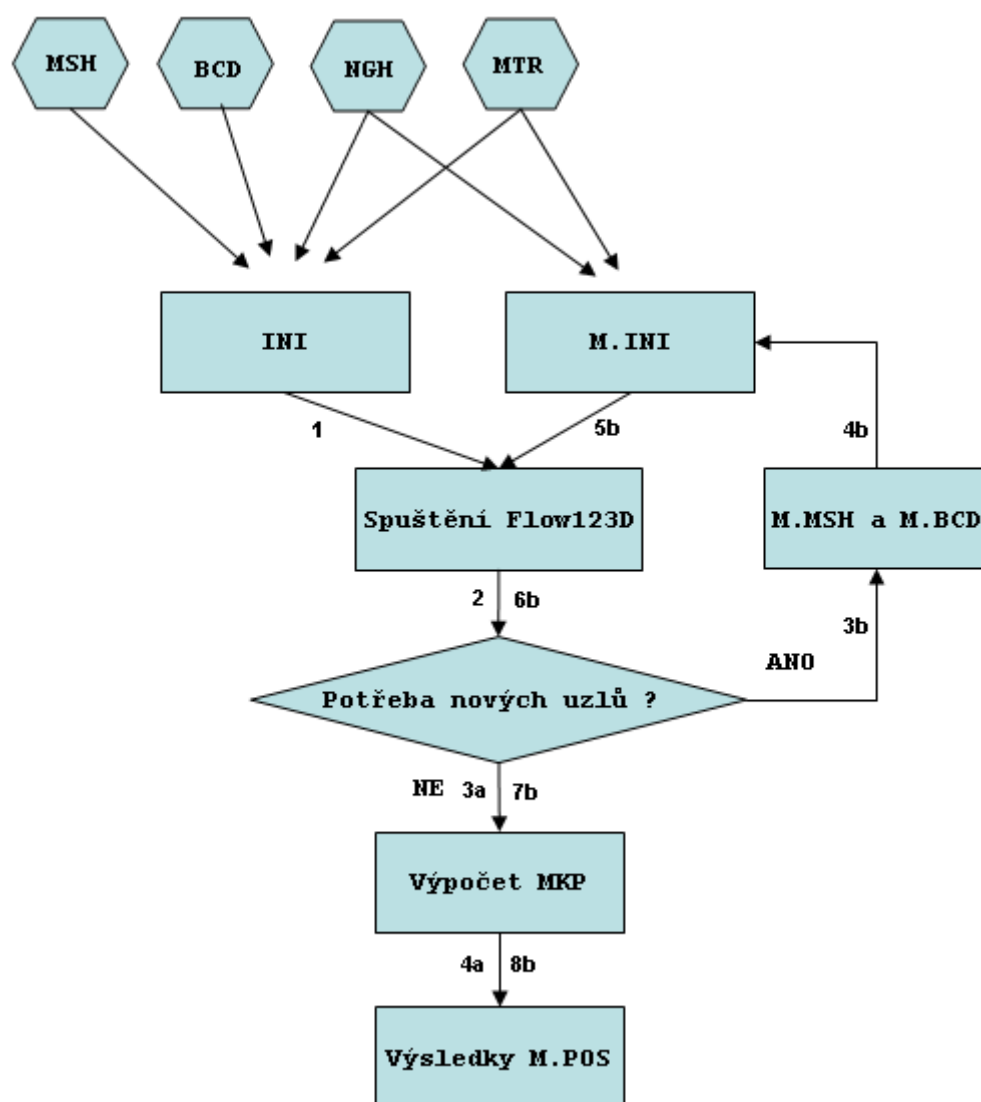
Hodnoty  $r$  jsou hodnoty pravé strany příslušného uzlu, které jsou součtem původní pravé strany a právě přetokem  $\mathbf{u}$ . Tím je hotov návrh komunikace 1D s 2D elementy stejně tak 2D s 3D elementy a můžeme přistoupit k popisu implementace těchto myšlenek do struktur programu Flow123D.

## 3 IMPLEMENTACE ALGORITMU

### 3.1 Použití primárního modelu

Podívejme se, jakým způsobem je náš algoritmus patrný z hlediska uživatele programu Flow123D.

Podle obr. 3-1 je před samotným spuštěním programu potřeba zadat do INI souboru názvy souborů sítě (\*.msh), okrajových podmínek (\*.bcd), sousedností (\*.ngh) a materiálů (\*.mtr). Poznamenejme ještě, že soubor sousedností dokáže Flow123D vytvořit také, ovšem za předpokladu, že zbývající tři soubory již máme vytvořené.



Obr. 3-1, vývojový diagram

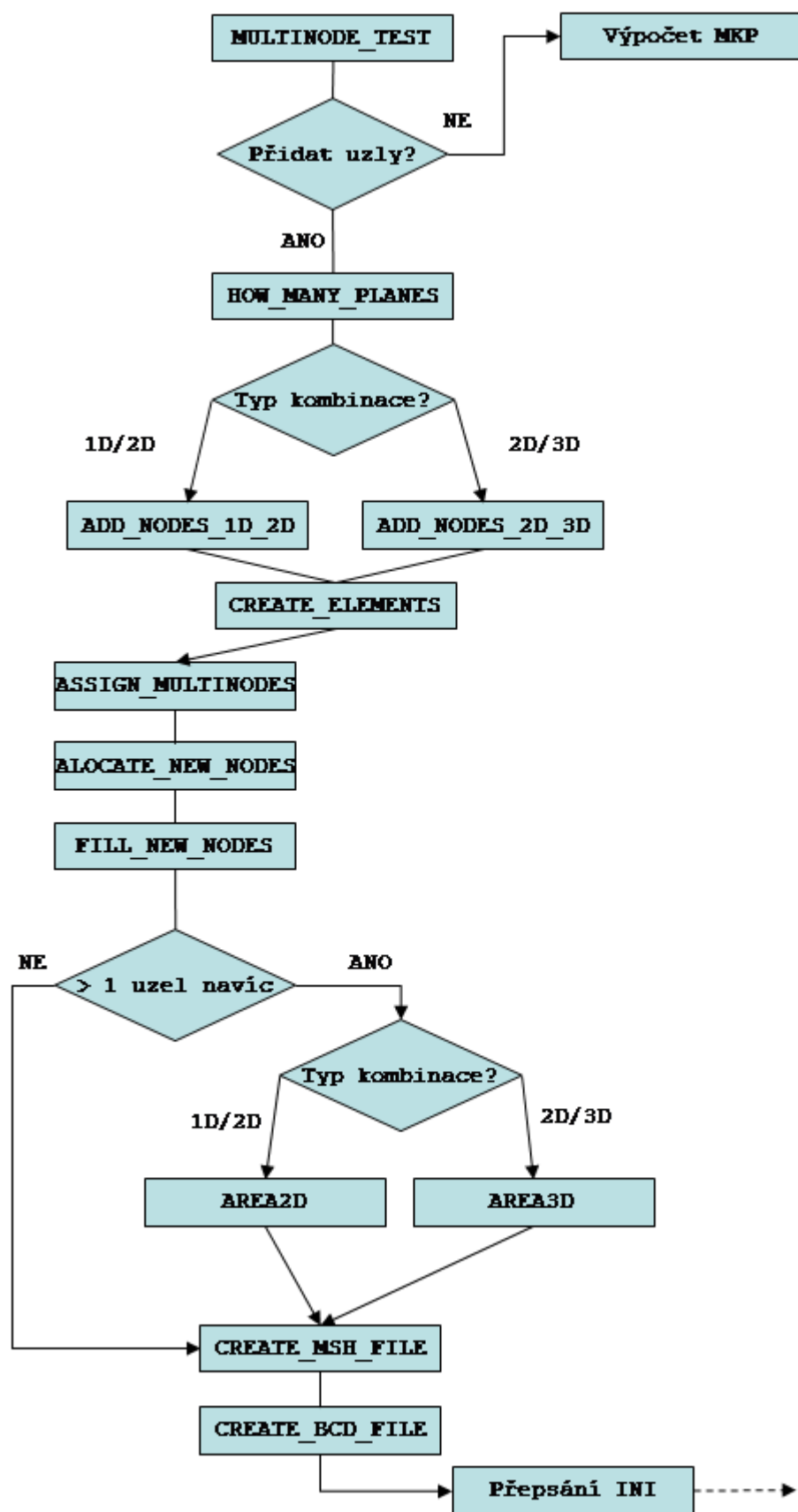
Pokud máme tyto soubory zadány do \*.ini souboru, je možné spustit program. Podrobný popis spuštění programu stejně jako popis všech potřebných souborů lze nalézt v příloze.

Pokud jsou v síti, na které se počítá, přítomny elementy různých dimenzí spojené společnými uzly, dojde ke spuštění algoritmu přidávání uzlů. Jeho struktura je vysvětlena níže. Výsledkem našeho kódu je vytvoření nové sítě s přidanými uzly uložené do nového souboru sítě „m.msh“. Zároveň je vytvořen i nový soubor okrajových podmínek „m.bcd“. Následně program automaticky přepíše v \*.ini soubor na „m.ini“ (z důvodu zálohy původního \*.ini souboru) a v něm název původního souboru sítě na „m.msh“ a stejně tak původní soubor okrajových podmínek na vytvořený „m.bcd“ a spustí program Flow123D znovu. Tentokrát dojde v výpočtu úlohy proudění, jejíž výsledek zapíše do souboru „m.pos“.

## **3.2 Algoritmus z hlediska programátora**

Na obr. 3-2 je naznačen chod programu tak, jak byl do programu Flow123D implementován. Následně si popíšeme každý krok tohoto algoritmu. Je koncipován do dvou fází. V první fázi je nejprve otestována síť, zda je nebo není třeba oddělovat elementy různých dimenzí, tuto fázi zobrazuje obr. 3-2. Pro případ, že je třeba oddělovat, vytvoří již zmíněné dva soubory „m.msh“ a „m.bcd“. Struktura obou souborů je uvedena v příloze této práce.

Ve druhé fázi na obr. 3-5 je již síť vyhodnocená jako způsobilá k výpočtu, ale je ještě potřeba zapsat správné hodnoty přestupových koeficientů na správné pozice v globální matici, jak je uvedeno v (35) resp. (41). Nejprve začneme s popisem první fáze.



Obr. 3-2, 1. fáze, vývojový diagram

### 3.2.1 Multinode\_test

Pro většinu celého algoritmu byl pro přehlednost vytvořen nový \*.c soubor nazvaný `multinode.c`. Obsah tohoto souboru nyní popíšeme. Po načtení vstupních souborů úlohy a provedení všech výpočtů týkajících se topologie sítě a elementů dochází v proceduře `multinode_test` k projití všech uzlů sítě a pokud je minimálně na jednom z nich připojen zároveň 2D a 1D nebo 2D a 3D element, pak je potřeba provést náš algoritmus.

V tomto případě je stále v této proceduře zjištěn počet uzlů, který je třeba přidat ke každému originálnímu uzlu pomocí procedury `Add_nodes_1D_2D` resp. `Add_nodes_2D_3D`. Tyto procedury využívají výstupu funkce `How_many_planes`. Ta vrací počet rovin napojených na daný uzel, počet elementů připojených k danému uzlu a výčet identifikačních čísel těchto elementů v každé rovině. Rovina je tvořena vždy 2D elementy. V pojetí 3D prostředí tedy rovina vyjadřuje puklinu a v pojetí 2D prostředí vyjadřuje libovolně natočenou rovinu porézního media.

### 3.2.2 Add\_nodes

Tyto procedury přidávají nové uzly k originálním uzlům tím způsobem, že pouze zvyšují hodnotu příznaku `added_nodes` ve struktuře každého uzlu, tím si každý uzel s sebou nese informaci o počtu potřebných uzlů. Počet nových potřebných uzlů z hlediska struktury sítě je pro kombinaci 1D a 2D elementů roven počtu liniových elementů ležících v jedné rovině a připojených k danému uzlu, výjimku tvoří pouze uzly, na které je připojena pouze jedna linie jako na obr. 2.4, pak je zřejmé, že se jedná o krajní linii (kraj pukliny) v oblasti a k tomuto uzlu přidáváme dva nové uzly z toho důvodu, aby puklina ani na jejím okraji nemohla být obtékána.

U kombinace 2D a 3D elementů je počet nových potřebných uzlů roven pro případ jedné roviny procházející daným uzlem dvěma přidaným uzlům, pro dvě roviny procházejícím tímto uzlem je roven čtyřem novým uzlům a pro tři roviny procházející tímto uzlem je roven osmi novým přidaným uzlům, více rovinám procházející jedním uzlem se tato práce nezabývá.

Dále jsou v rámci `multinode_test` volány i procedury `Create_elements`, `Create_msh_file` a `Create_bcd_file`. V případě, že není třeba vytvářet nové uzly (na žádném z uzlů sítě není ani jeden uzel společný pro 1D a 2D elementy resp. 2D a 3D elementy), je procedura ukončena a pokračuje se ve výpočtu běžným způsobem.

### 3.2.3 Create\_elements

Úkolem této procedury je zavést nové číslování uzlů **assign\_multinodes**, alokovat v paměti místo pro nové uzly elementů **allocate\_new\_nodes** a přiřadit tyto nové uzly elementům **fill\_new\_nodes**.

Po přidání nových uzlů každému originálnímu uzlu vyvstala otázka, jak přistoupit k očíslování nových uzlů v novém m.msh souboru. Nabízely se dvě varianty. První byla původní očíslované uzly ponechat a očíslovat pouze nové uzly a druhá byla číslovat všechny postupně od nuly, což by vedlo k tomu, že by uzly se stejnou polohou byly hodnotami identifikačních čísel u sebe. Zvolili jsme z důvodu přehlednosti a snazšího dohledávání v rozsáhlejších sítích druhou variantu, přechíslovat vše od nuly.

### 3.2.4 Assign\_multinodes

V této proceduře se alokují místa v paměti pro novou celočíselnou proměnnou *node\_number* ve struktuře uzlu, do které se zapíše jeho nové identifikační číslo. Počet alokovaných míst je roven počtu potřebných uzlů k přidání (*added\_nodes*) + originální uzel, který je znásobován. Takže uzel, ke kterému není třeba přidávat žádný násobný uzel, bude mít pouze jednu proměnnou *node\_number*, ve které bude uloženo jeho nové id. číslo.

### 3.2.5 Allocate\_new\_nodes

V této proceduře se již zabýváme elementy, alokujeme zde u každého elementu sítě celočíselné proměnné *element\_node*. Jejich počet je roven počtu uzlů daného elementu (linie dva, trojúhelník tři, čtyřstěn čtyři). Do této sady proměnných je třeba zapsat identifikační čísla nových uzlů, které tvoří daný element. Tím lze pak jednoduše vytvořit potřebný m.msh soubor.

### 3.2.6 Fill\_new\_nodes

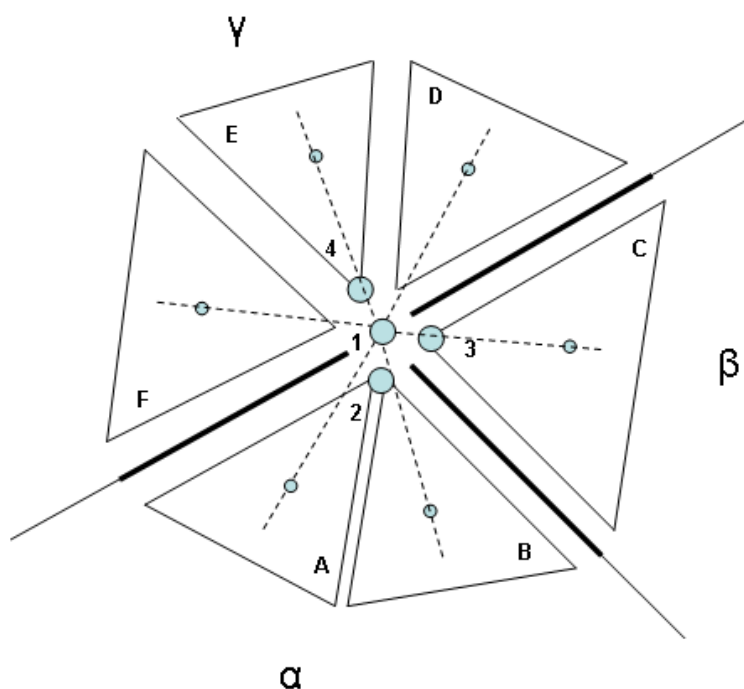
Poslední procedura je nejrozsáhlejší a nejsložitější z celého algoritmu, protože je zde nutné přiřadit ty správné uzly k jednotlivým elementům (neboli naplnit v předešlé proceduře vytvořené sady proměnných *element\_node* jednotlivých elementů). Prochází se uzel po uzlu v celé síti a testuje se, zda je daný uzel znásobený. Nastávají dvě možnosti výsledku testu. Buď je daný uzel samotný, pak je rovnou přiřazen elementům,



které tento uzel tvoří. Nebo má k sobě tento originální uzel ještě další nové přidané uzly a pak je třeba pomocného algoritmu, díky kterému se přiřadí nový uzel vhodným elementům, na obr. 3-2 je tato podmínka vyjádřena podmínkou, zda je u originálního uzlu více jak jeden přidaný uzel. Nesmí tak dojít k situaci, kdy je v síti přiřazen uzel např. 1D elementu a současně 2D elementu.

### 3.2.7 Area2D

Tato procedura byla vytvořena pro znásobené uzly kombinace 1D-2D. Názorně algoritmus popíšeme podle obr. 3-3. V případě, že je k jednomu originálnímu uzlu připojen více jak jeden 1D element, v našem případě máme k jednomu uzlu připojeny tři liniové elementy (tučně zvýrazněny), které tvoří tři vektory. Je vybrán jeden 1D element a k němu druhý 1D element, který je k němu ve smyslu velikosti úhlu pootočení nejblíže, nejprve máme úhel. V této výseči je přiřazen jeden nový uzel jen těm elementům, které mají spojnici středu elementu s originálním uzlem ležící právě v této výseči. Pro výseč úhlu  $\alpha$  to jsou elementy A a B a je k nim přiřazen uzel 2. Dále algoritmus pokračuje od poslední vybrané linie (1D elementu) a opět k sobě najde nejbližší ještě nepoužitou linii na obrázku se jedná o výseč o velikosti  $\beta$ , tímto způsobem je pokryta celá plocha kolem uzlu a přiřazeny správné uzly ke správným elementům.

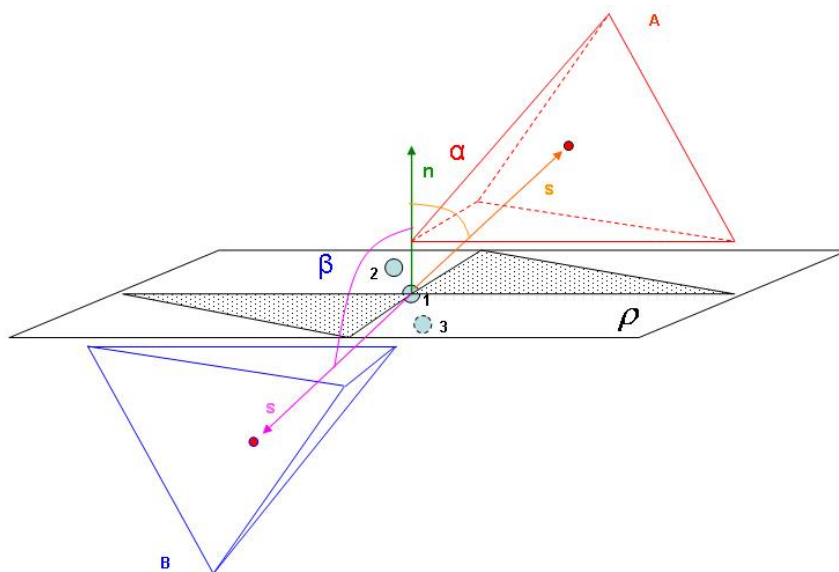


Obr. 3-3, Area2D

### 3.2.8 Area3D

Pro kombinaci 2D-3D je pro znásobené uzly vytvořena procedura **Area3D**. Pro případ, že daným uzlem prochází pouze jedna puklina = jedna rovina, tato rovina rozděljuje oblast kolem uzlu na dvě části, jsou tedy potřeba dva nové uzly (původní je ponechán puklině = 2D elementu), tyto dva nové uzly už má původní uzel uchovány v paměti z procedury **assign\_multinodes**.

Pokusíme se algoritmus popsat pomocí obr. 3-4, na něm je naznačena puklina jako rovina  $\rho$ , ta je tvořena 2D elementy. Zaměříme se na střed roviny kde leží uzel 1, který náleží 2D elementům v puklině. Jeden přidáný uzel je přiřazen jednomu poloprostoru např. uzel 2 hornímu poloprostoru, kterou puklina vytyčí, a uzel 3 ke spodnímu poloprostoru. Je třeba přiřadit 3D elementy přidáním uzlům ležícím v daném poloprostoru, pro jednoduchost máme na obrázku pouze dva elementy A a B. Element A leží nad puklinou a element B leží pod puklinou, nemůžou tedy mít společné uzly. Proto je v této proceduře porovnáván úhel mezi normálovým vektorem roviny (pukliny), značen  $\mathbf{n}$ , se spojnicí daného uzlu (uzel 1) se středem elementu A, značen vektorem  $\mathbf{s}$ . Pokud je tento úhel menší než 90 stupňů, pak přiřadíme vybraný element A vybranému novému uzlu 2. Vezmeme-li případ elementu B, úhel mezi normálovým vektorem  $\mathbf{n}$  a vektorem  $\mathbf{s}$  je větší než 90 stupňů, a proto nedojde k přiřazení uzlu 2 elementu B. Po projití všech elementů připojených k danému uzlu se otočí směr normálového vektoru a to samé se provede znovu. Výsledkem tohoto postupu je správné přiřazení nových uzlů k 3D elementům ležícím v daném poloprostoru. Pro případy dvou či třech rovin (puklin) procházejících jedním uzlem je postup analogický.



Obr. 3-4, Area3D

### 3.2.9 Create\_msh\_file

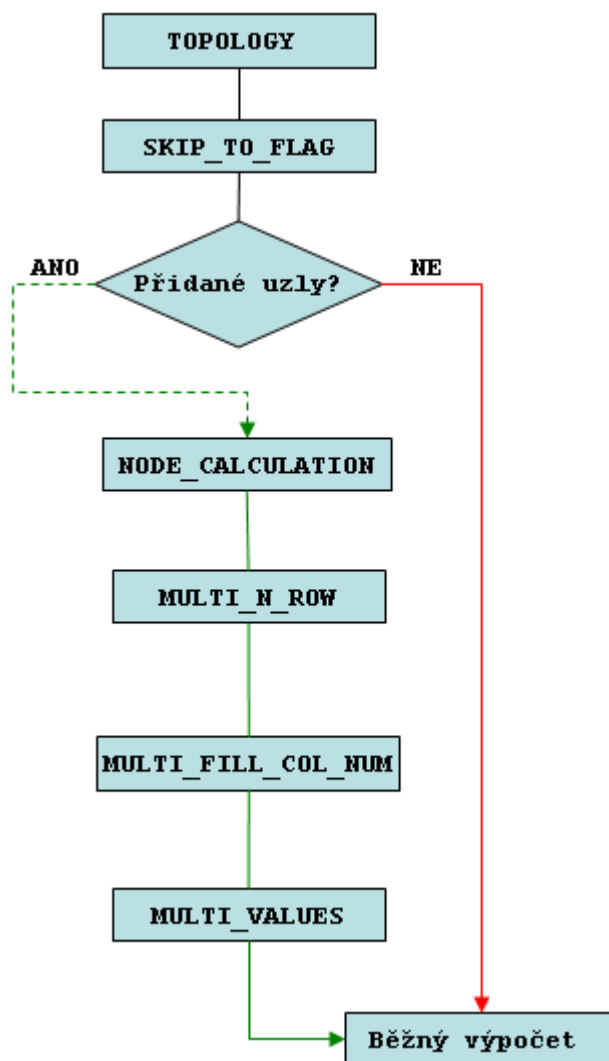
Po provedení procedury **fill\_new\_nodes** již máme ke každému elementu přiřazeny uzly, které jej tvoří. Dále již lze vytvořit nový m.msh soubor pomocí procedury **create\_msh\_file**. Do něj jsou vypsány postupně všechny uzly se svými souřadnicemi, elementy s materiály a uzly, které je tvoří. Nakonec je na konci souboru vytvořena značka „m“, v programu znamená, že se jedná o soubor s násobnými uzly potřebná pro další správný chod programu ve zmíněné druhé fázi.

### 3.2.10 Create\_bcd\_file

A jako poslední v celém souboru je vytvořen nový \*.bcd soubor procedurou **create\_bcd\_file**. Do něj jsou vepsány všechny předepsané okrajové podmínky stejné jako jsou v originálním souboru, jen s aktuálními nově očíslovanými uzly.

### 3.2.11 Topology

Tím je popsán celý soubor multinode.c a zbývá popsat druhou fázi našeho algoritmu obr. 3-5. Nejprve je již v části **Topology** přečten \*.msh soubor počítané úlohy a je zjištěno (procedura **skip\_to\_flag**) pomocí přítomné značky „m“ na konci souboru, že tato síť je přepracována na náš způsob komunikace elementů různých dimenzí a tím je nastaven globální příznak *multinodes* (ve struktuře Mesh) na hodnotu 1. Dále program běží standardně až do části **node\_calculation**, kde se mj. pro každý uzel nastavují do proměnných *n\_row*, *col\_number* a *value*. Po řadě znamenají počet nenulových hodnot v řádku a číslo sloupce globální matice a hodnoty koeficientů z lokálních matic elementů na této pozici. Stávající algoritmus plnění výše uvedených proměnných danými hodnotami je závislý na informacích jako jsou počty a typy elementů připojených k jednotlivým uzlům. Tyto údaje jsou jednotlivým uzlům přiřazovány již ve zmíněné proceduře **Topology**. Problém je v tom, že po načtení námi vytvořené sítě jsou elementy různých dimenzí fyzicky odděleny tzn. nemají společné uzly a nemohou o sobě „vědět“, ačkoliv jsou přímými sousedy. My však potřebujeme pro naplnění globální matice výše odvozenými koeficienty přestupu pro jednoznačné určení pozice daného koeficientu v matici počet nenul v řádku a sloupec matice, proto jsou do procedury **node\_calculation** dodefinovány procedury **Multi\_n\_row**, **Multi\_fill\_col\_num** a **Multi\_value**.



Obr. 3-5, 2.fáze, vývojový diagram

### 3.2.12 Multi\_n\_row

V této proceduře je pro daný násobný uzel zvětšena hodnota  $n\_row$  o hodnotu, která je rovna počtu uzlů sousedícího elementu, ale majícího identifikační číslo větší než je id. číslo daného uzlu (poslední podmínka vyplývá z vynechávání hodnot pod diagonálou u symetrické matice).

### 3.2.13 Multi\_fill\_col\_num

Procedura zajišťuje přiřazení správné hodnoty sloupce globální matice k danému násobnému uzlu, kde hodnota sloupce je rovna identifikačnímu číslu uzlu sousedního elementu.

### 3.2.14 Multi\_values

Poslední operace celého algoritmu je již zmiňované vkládání hodnot nových uzlů do globální matice, provádí se v proceduře **multi\_values** a jednoduše přičítá resp. odečítá hodnoty, jak je znázorněno v maticích (35) a (41). Hodnoty přestupových koeficientů jsou zjišťovány z \*.ngh souboru, kde je každé sousednosti elementů různých dimenzí (typ sousednosti 20) tento koeficient přiřazen. Po vytvoření \*.ngh souboru programem Flow123D je hodnota přestupových koeficientů nastavena implicitně na 1.0.

Dále je, již jako ve standardním výpočtu primární formulace MKP, vytvořena rovnice (26), předložena řešiči – Matlabu a vypočteny hodnoty tlakových výšek pro každý uzel a v postprocesu dopočteny vektory rychlostí proudění pro každý element sítě.

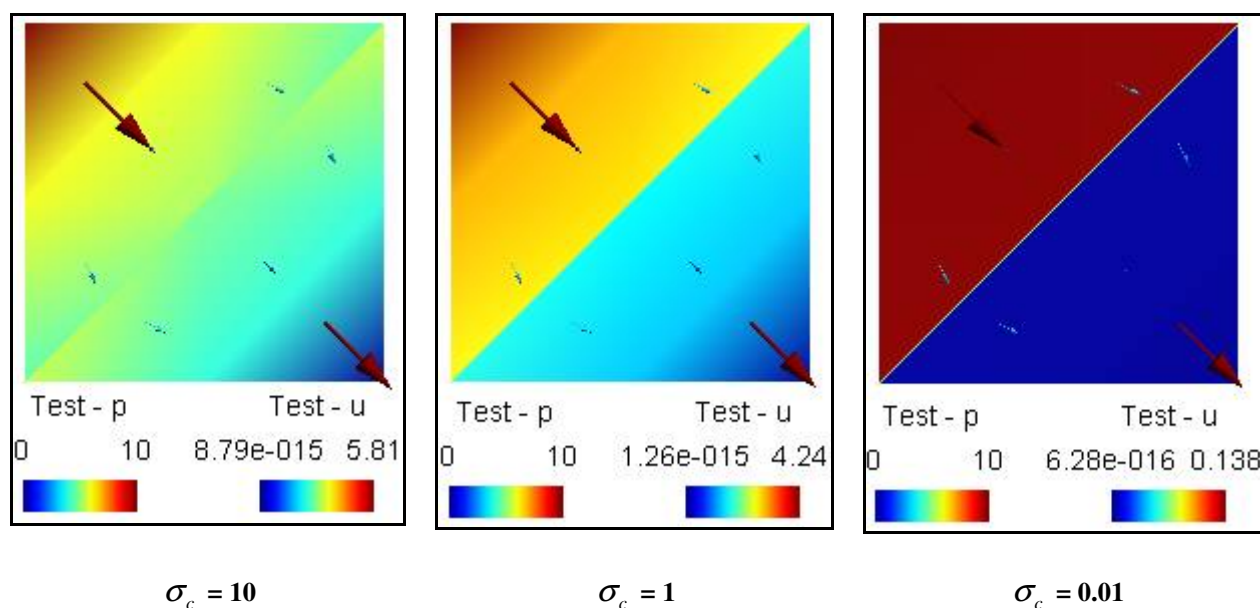
## 4 TESTOVACÍ ÚLOHY

Pro odladění a otestování správné funkčnosti implementovaného modelu je třeba použít jednoduchých testovacích úloh. Popisy úloh byly vytvořeny ručně, nebo s použitím generátoru puklinových sítí vyvinutým speciálně pro Flow123D. Globální matici sestavil program Flow123D, k vyřešení byl použit program MATLAB a samotné vykreslení výsledků bylo zobrazeno programem GMSH. Všechny zde představené testovací úlohy, program Flow123D i vykreslovací program GMSH lze nalézt v příloženém CD.

### 4.1 Test z úvodu

Vezměme znovu úlohu z úvodu, na které byla ukázána nezávislost toku oblastí na propustnosti pukliny, což bylo označeno za chybné chování modelu. Předpokládejme stejné okrajové podmínky, tlaková výška 10m v levém horním rohu, tlaková výška 0m v pravém dolním rohu, ostatní uzly na okraji oblasti jsou automaticky nastaveny na nepropustné. Dále předpokládáme stejnou propustnost porézního média (2D element) a propustnost pukliny (1D element) nastavíme na konstantní hodnotu 1.0, měnit budeme pouze přestupový koeficient  $\sigma_c$  pro sousednosti 1D a 2D. Tento koeficient je pro všechny tyto sousednosti vždy stejný i v dalších testech. Síť je shodná se sítí v úvodu na obr. 1-1.

**Výpočet tlakových polí a vektorů rychlostí:**



Obr. 4-2, síť

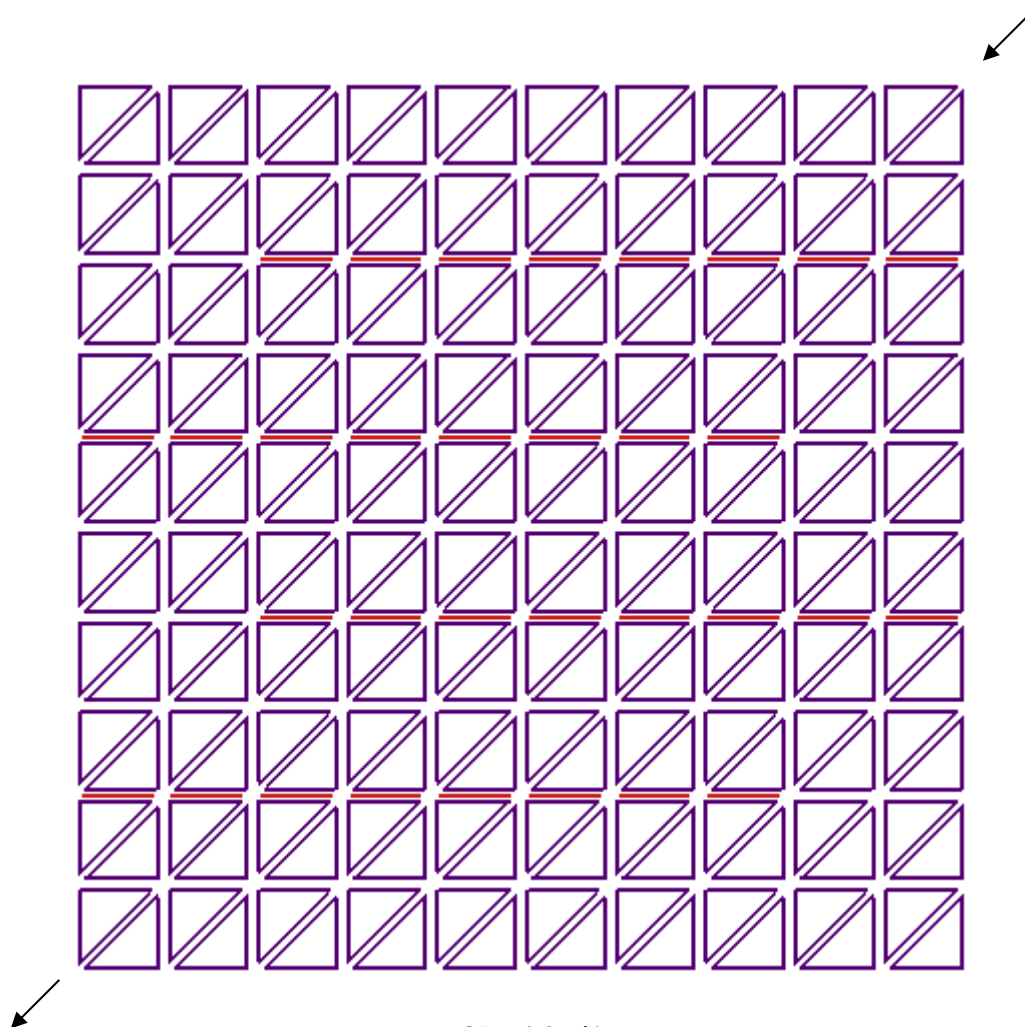
### Výsledek testu:

Z uvedených výsledků na obr. 4-2 je vidět závislost rychlosti toku oblastí na propustnosti pukliny nastavené hodnotou koeficientem přestupu mezi porézním médiem a puklinou. S tímto výsledkem jsme spokojeni, neboť již dokážeme počítat s oblastmi, ve kterých tok směřuje napříč přes puklinu.

## 4.2 Test 2D-1D na rozsáhlejší oblasti

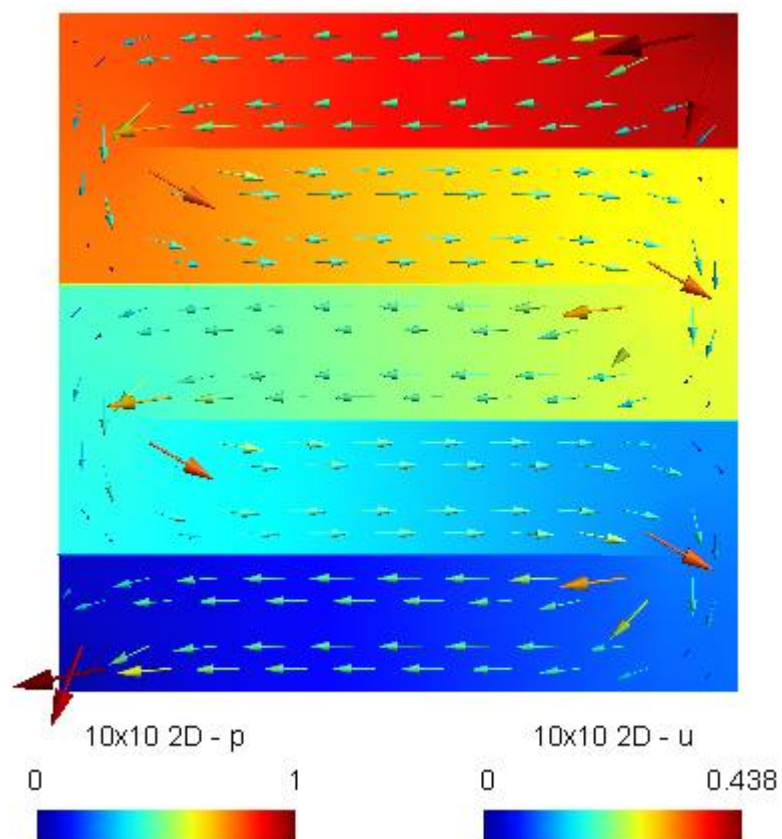
V tomto testu budeme počítat na rozsáhlejší oblasti než je tomu v prvním testu.

Uvažujme síť na obr. 4-3, v porézním médiu o propustnosti  $K = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  se nacházejí pukliny (vyznačeny červenými úsečkami) všechny se stejnou propustností  $K = 1$ . Opět budeme měnit pouze přestupový koeficient  $\sigma_c$  definovaný každé sousednosti elementů 1D a 2D. Tok je nastaven z pravého horního rohu (tlaková výška v tomto uzlu nastavena na hodnotu 1m) do levého dolního (tlaková výška nastavena na hodnotu 0m).

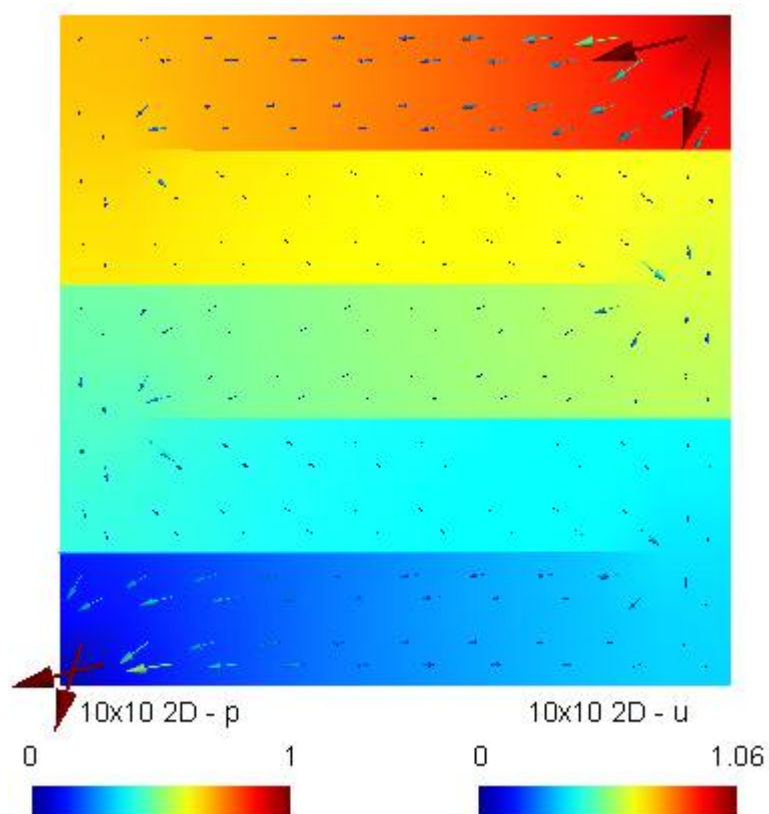


Obr. 4-3, síť

**Výpočet tlakových polí a vektorů rychlostí:**

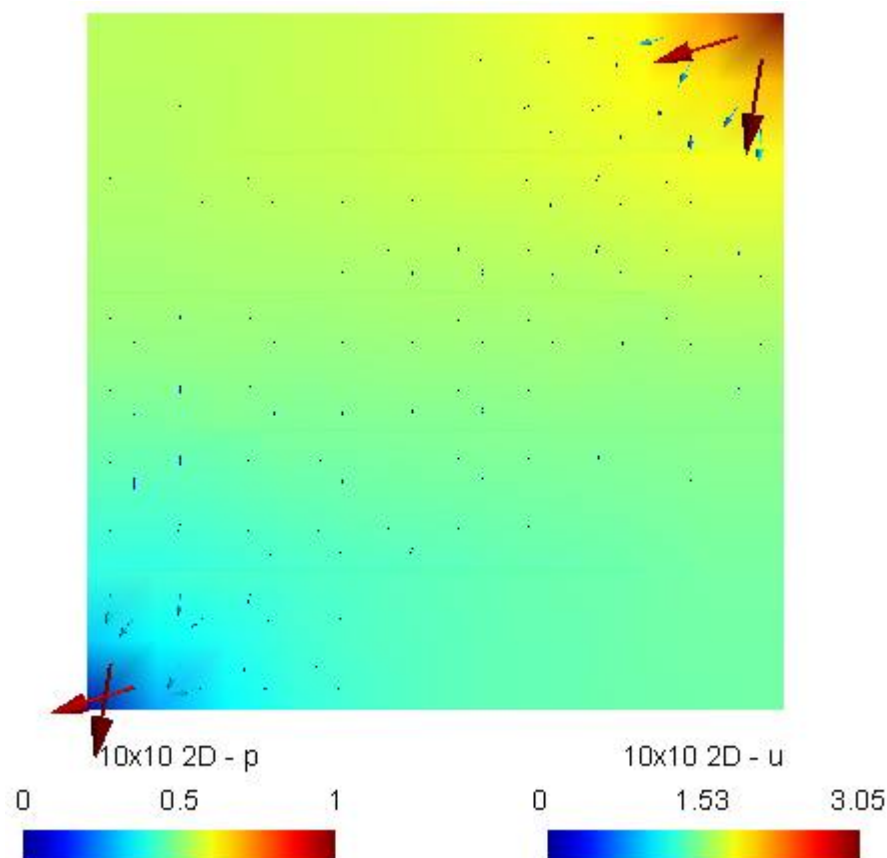


**Obr. 4-4,  $\delta = 0.01$**



**Obr. 4-5,  $\delta = 1.0$**





Obr. 4-6,  $\sigma_c = 10.0$

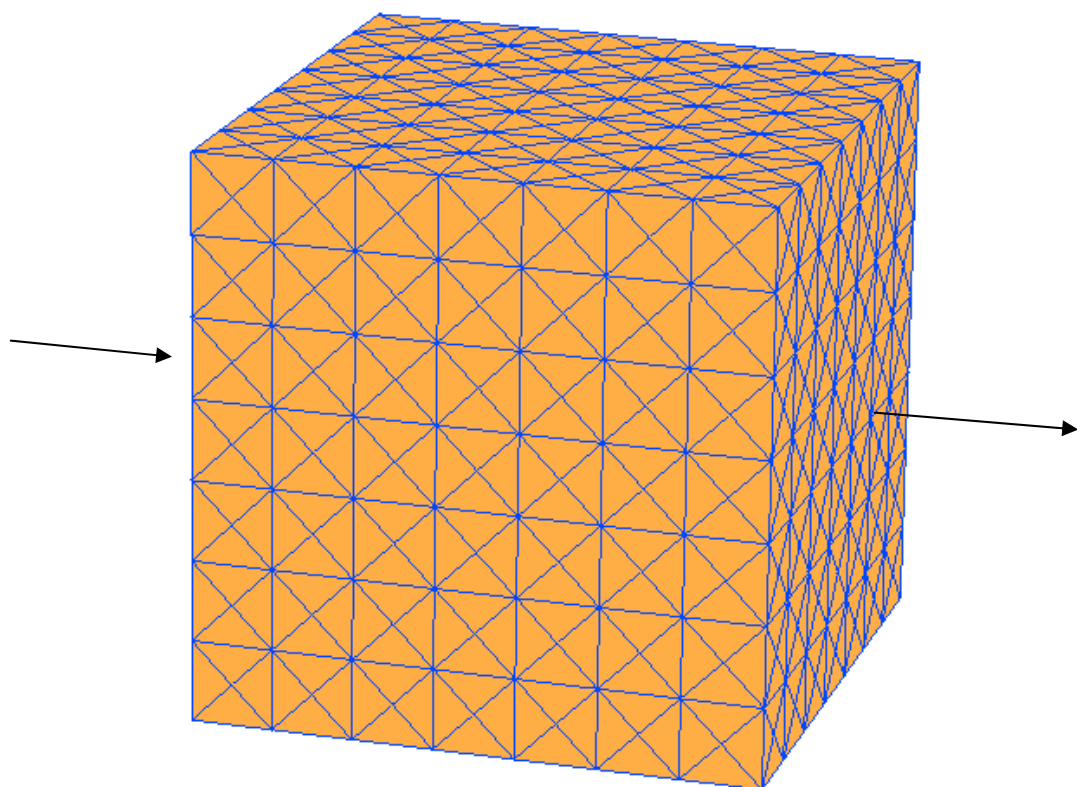
#### Výsledek testu:

Na této modelové úloze je ukázáno, jak vypadá tlakové a rychlostní pole v případě, že puklina se chová jako hydraulický izolátor. Pro tento případ je  $\sigma_c = 0.01$  a z příslušného obr. 4-4 je zřejmé, že podzemní voda obtéká puklinové pole bez známek prostupu vody přes pukliny. Pro případ, kdy  $\sigma_c = 10.0$  je patrné, že v tlakovém a v rychlostním poli jsou pukliny maximálně propustné, chovají se jako hydraulický kolektor.

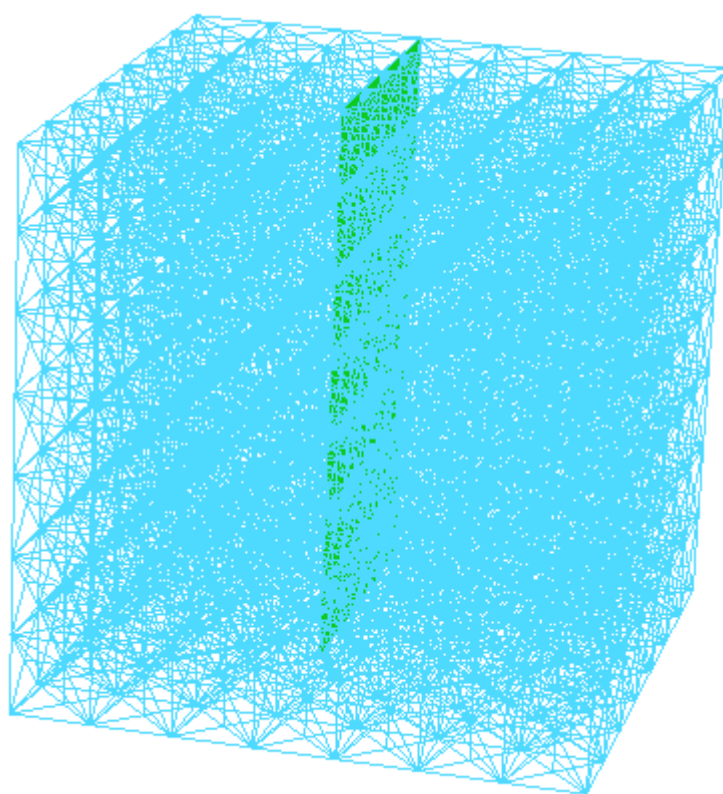
### 4.3 Test 2D-3D na rozsáhlejší oblasti

Nyní budeme testovat případ, kdy máme porézní médium vyjádřeno 3D elementy (čtyřstěny) a pukliny jsou vyjádřeny 2D elementy (trojúhelníky). Na obr. 4-7 je představena síť ve tvaru krychle, která má uprostřed vloženu jednu puklinu obr. 4-8. Tok v této oblasti je nastaven od levého boku krychle k pravému boku, naznačeno šipkami, tomu odpovídající tlaková výška ve všech uzlech levého boku krychle je na hodnotě 2m a na všech uzlech pravého boku je na hodnotě 1m, všechny ostatní hraniční uzly jsou opět nastaveny automaticky na nepropustné. Identickou úlohu jsme nechali spočítat pro porovnání modelem založeným na smíšené hybridní formulaci MKP, první

obrázek je vždy pro dané nastavení koeficientu propustnosti náš model a spodní obrázek je smíšený hybridní model (označen MH).

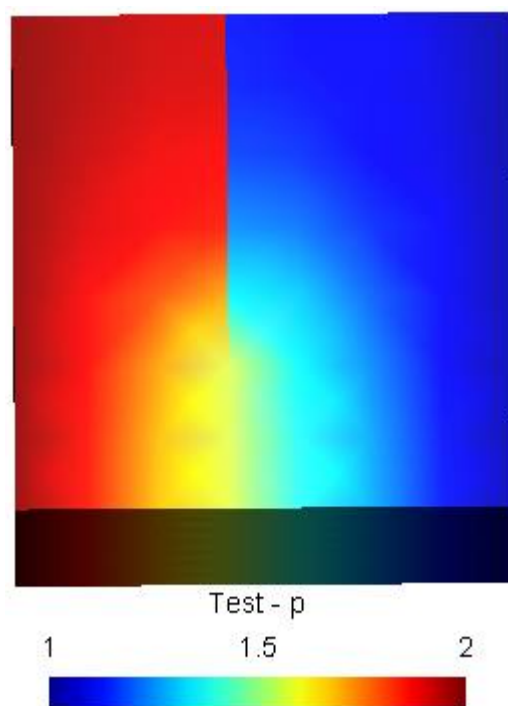


**Obr. 4-7, síť**

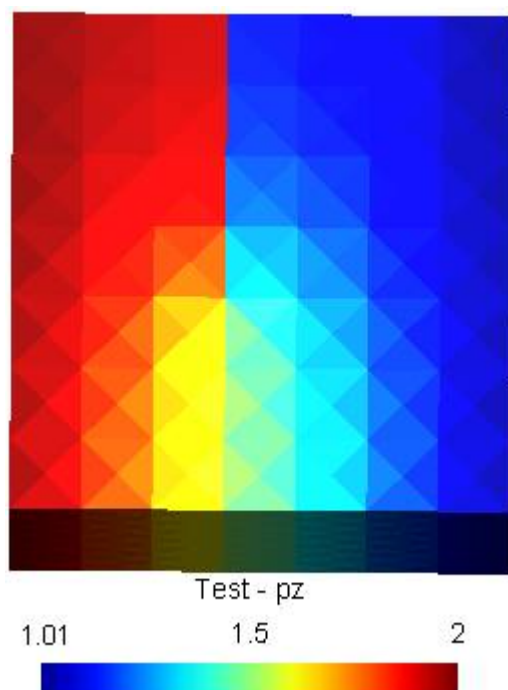


**Obr. 4-8, puklina v síti**

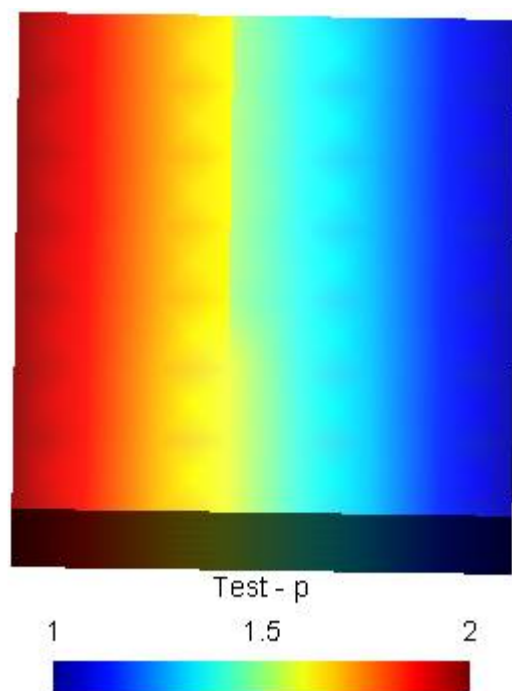
**Výpočet tlakových polí (pohledy na horní stěnu krychle):**



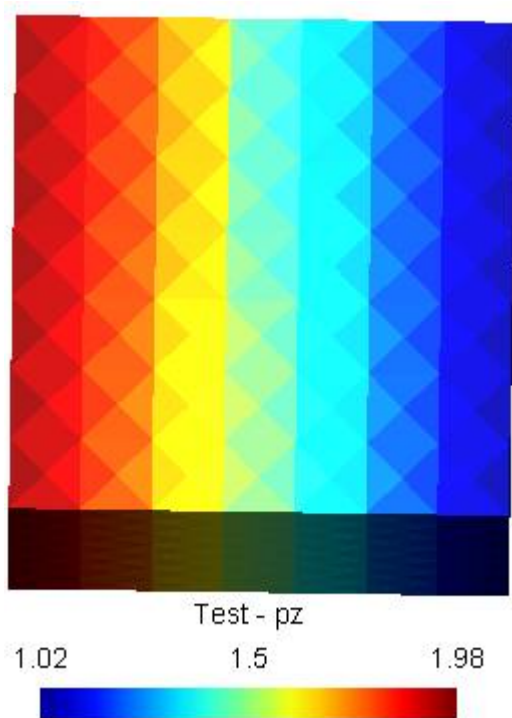
**Obr. 4-9,  $\delta = 0.001$ , Primar**



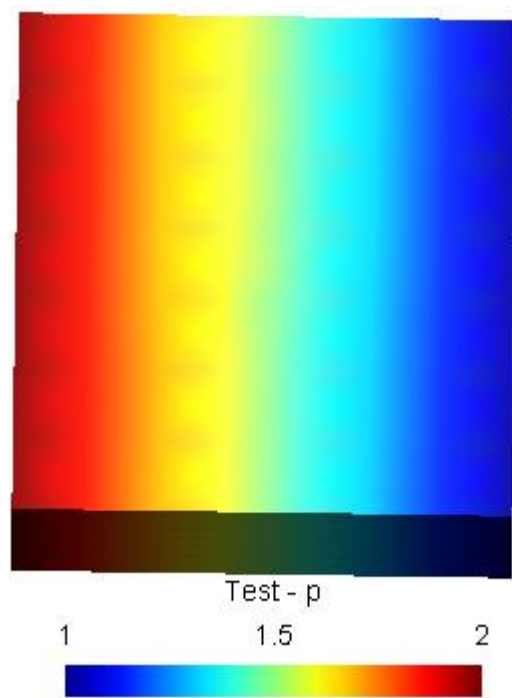
**Obr. 4-10,  $\delta = 0.001$ , MH**



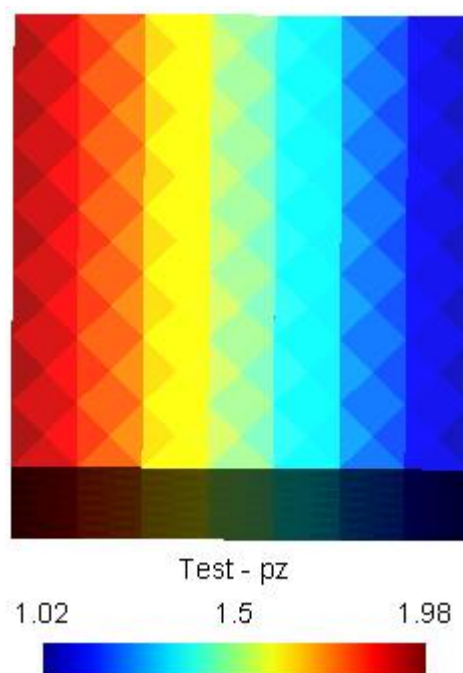
Obr. 4-11,  $\delta = 1.0$ , Primar



Obr. 4-12,  $\delta = 1.0$ , MH



Obr. 4-13,  $\delta = 10.0$ , Primar

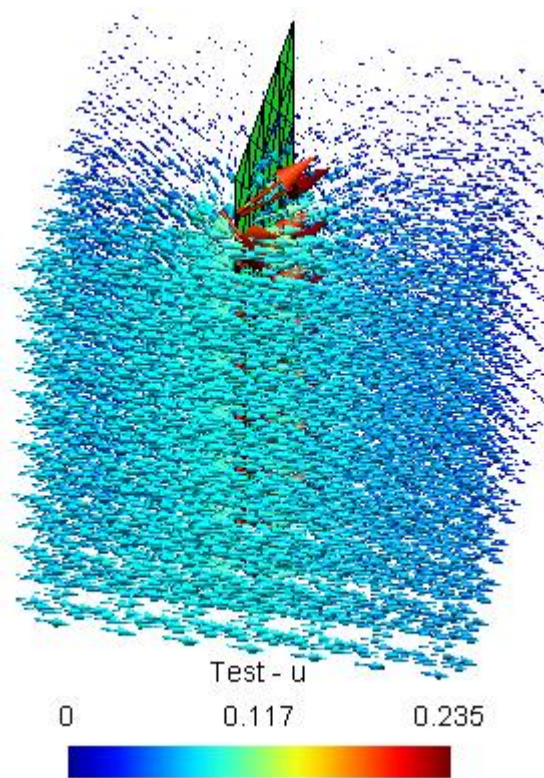


Obr. 4-14,  $\delta = 10.0$ , MH

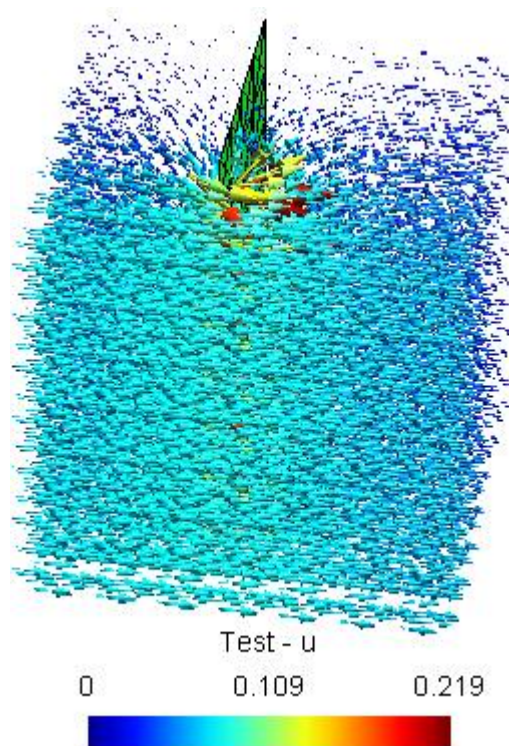
### Výsledky testu:

Z tlakových polí můžeme vidět při nastaveném přestupovém koeficientu na hodnotu 0.001, že puklina rozdělí tlakové pole. Naopak při hodnotě přestupového koeficientu nastaveného na 10.0 je patrné, že puklina se v tlakovém poli vůbec neprojeví. Odchylny krajních hodnot tlakových výšek MH modelu jsou způsobeny nutným přepočtem okrajových podmínek z uzlů do těžišť hraničních stěn čtyřstěnů. Jinak je patrné, že oba modely se v rozložení tlakových výšek shodují.

**Výpočet rychlostních polí:**

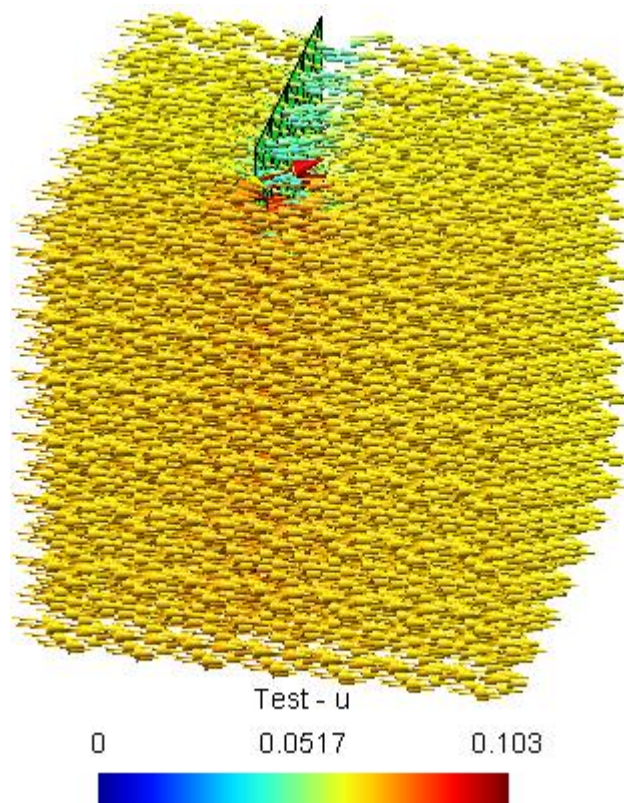


**Obr. 4-15,  $\delta = 0.001$ , Primar**

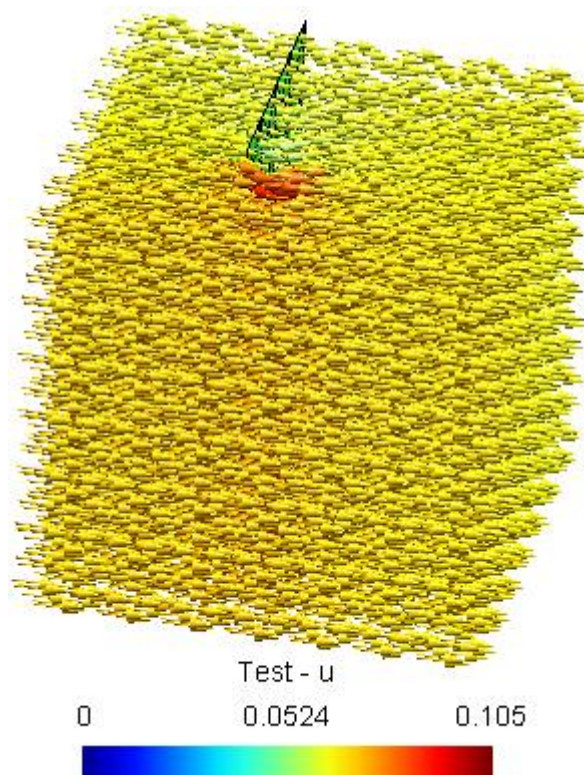


**Obr. 4-16,  $\delta = 0.001$ , MH**

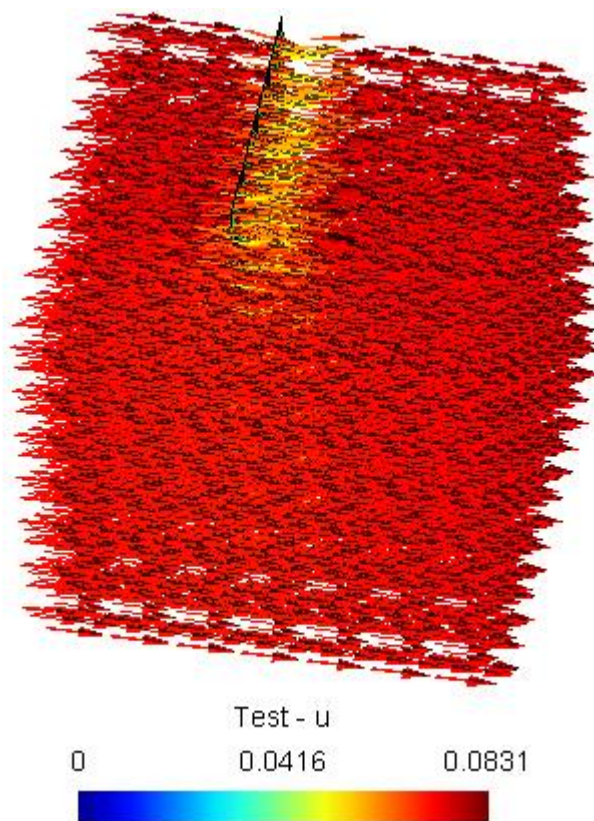




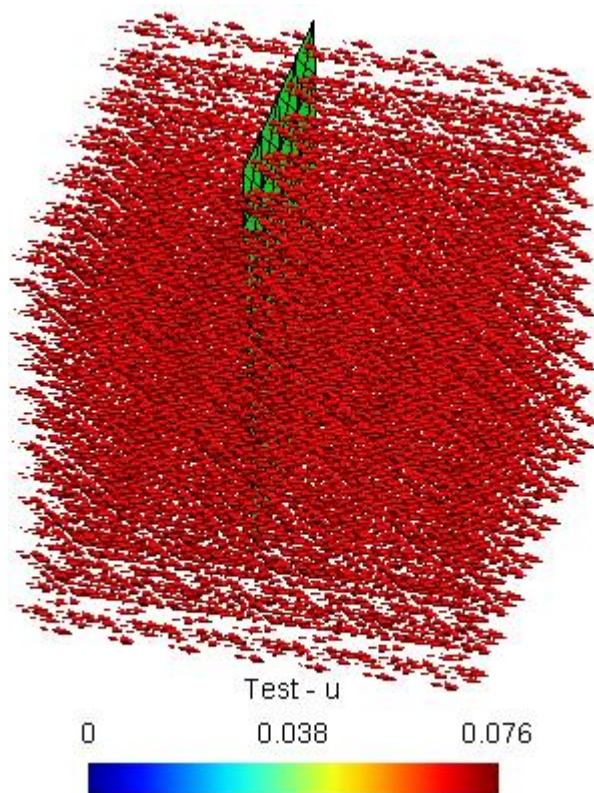
**Obr. 4-17,  $\delta = 1.0$ , Primar**



**Obr. 4-18,  $\delta = 1.0$ , MH**



**Obr. 4-19,  $\delta = 10.0$ , Primar**



**Obr. 4-20,  $\delta = 10.0$ , MH**



### Výsledek testu:

Z výsledků rychlostních polí lze pro kombinaci 2D-3D elementů usoudit správnou funkčnost i pro tuto kombinaci. Pro přestupový koeficient  $\sigma_c=0.001$  se opět puklina chová jako hydraulický izolátor a naopak pro  $\sigma_c=10.0$  jako hydraulický kolektor.

Viditelná ztráta rychlosti proudění obr. 4-19 na obou stranách pukliny při vysoké hodnotě přestupového koeficientu je způsobena přetokem z těžiště 3D elementu do těžiště 2D elementu a stejně tak přetokem z těžiště 2D elementu do těžiště 3D elementu na druhé straně pukliny. Z tohoto důvodu by bylo lepší použít přetokový vztah ve tvaru:

$$\mathbf{u} = \sigma_c \left( \frac{p_1'' + p_2''}{2} - \frac{p_1' + p_2'}{2} \right) \quad (42)$$

ten zajišťuje přetok pouze z přilehlé stěny 3D elementu na 2D element. Přetok je tímto vztahem dán rozdílem tlaků na geometricky shodných útvarech.

## 5 ZÁVĚR

Tato diplomová práce úspěšně vyřešila v úvodu vytyčené problémy, byl navržen algoritmus pro aproximaci úlohy filtračního proudění pomocí primární formulace MKP a pro přidávání uzlů sítě na styku elementů různých dimenzí a pro stanovení přetoku mezi takovými uzly. Dále byl tento algoritmus implementován do programu Flow123D. Tato implementace byla posléze otestována na modelových úlohách, ve kterých výsledky testů prokázaly správnou funkčnost modelu.

Nyní lze s dobrými výsledky modelovat pomocí primární formulace MKP proudění podzemní vody v porézním médiu proloženým systémem puklin bez ohledu na orientaci jednotlivých puklin vůči směru toku oblastí. Díky v této práci navrženému přestupovému koeficientu lze nastavovat různé propustnosti libovolných puklin v oblasti a tím docílit toho, aby se v krajních případech puklina jevila jako hydraulický izolátor nebo naopak hydraulický kolektor.

Jako vhodnější se na základě výsledku testu v kapitole 4.3 předpokládá použít pro přetok mezi elementy různých dimenzí vztah (42), kde by se jednalo místo komunikace mezi dvěma elementy o komunikaci mezi elementem nižší dimenze a přilehlou stěnou elementu vyšší dimenze.

Jako možné pokračování této práce lze rozšířit algoritmus přidávání uzlů o složitější případy, kdy jedním uzlem prochází čtyři a více puklin, dále lze doporučit implementovat přetok mezi elementy různých dimenzí ve tvaru vztahu (42).

# Literatura

- [1] **Eriksson K., Estep D., Hansbo P., Johnson C.:** *Computational Differential Equations*. Cambridge University Press, 1996.
- [2] *Dokumentace a zdrojové kódy programu 123flow*. TU Liberec, KMO, 2005.
- [3] **Petr Šaloun:** *Programovací jazyk C pro zelenáče*. Neocortex, 1999.
- [4] **Feynman, Leighton, Sands:** *Feynmanovy přednášky 2/3*. Fragment, 2001.
- [5] **Severýn O.:** *Model filtračního proudění podzemní vody založený na smíšené hybridní formulaci*. Diplomová práce, TU Liberec, 1997.
- [6] **Palek J.:** *Model filtračního proudění podzemních vod založený na kombinovaném přístupu a primární formulaci MKP*. Bakalářská práce, TU Liberec, 2006.

## Příloha

V této části je popsána struktura přiloženého CD, postup spuštění programu Flow123D a stručný popis vstupních souborů do programu.

### **Struktura CD:**

Flow123D – obsahuje zdrojové kódy, exe – soubor programu a potřebné knihovny

Testovací úlohy – obsahem složky jsou testovací úlohy

GMSH – obsahuje program pro vykreslení výsledků (\*.pos)

Diplomová práce.pdf

### **Spuštění programu Flow123D:**

Program 123Flow je konzolová aplikace v prostředí win32 a je spuštěn příkazovou řádkou ve tvaru:

**jméno\_programu\_ -s\_ jméno\_řídícího\_souboru**

příklad: flow123d.exe -s kanal.ini

### **Poznámka:**

Pro jednoduchost je k programu Flow123D.exe přiložen i **f.bat** soubor, který má již výše uvedený spouštěcí příkaz přednastavený, ke spuštění programu Flow123D stačí jen spustit tento soubor **f.bat**. Příkaz, který by se psal jinak do příkazové řádky, lze napsat jednou v editaci souboru **f.bat** a dále lze již stále spouštět jen tento soubor.

Povinný parametr *jméno\_řídícího\_souboru* určuje plné jméno a cestu k řídicímu souboru pro běh programu. V případě výpočtu založeného na primární formulaci se předpokládá nainstalovaný **Matlab**. Pro bezproblémové spuštění je třeba z adresáře **Flow123D/Program + knihovny** do adresáře obsahující řídicí soubor \*.ini požadované úlohy nakopírovat:

**Soubor\_flow123d.exe**

**Dforrt.dll**

**gm6.dll**

**gm6.lib**

Pro vykreslení výsledků je vhodný přiložený program GMSH, výsledky jsou zapsány v souboru \*.pos, který je vytvořen ve stejném adresáři, kde se nachází řídicí soubor.

## Vstupní a výstupní soubory

### Řídící soubor

Soubor má strukturu standardního ini-souboru používaného v prostředí MS Windows, tj. sekce, klíče a jejich hodnoty. Pro zápis čísel platí konvence jazyka C. Jména souborů mohou být zapsána buď jako absolutní cesty, nebo relativně vzhledem k adresáři, ze kterého je program spouštěn. Jména souborů se zadávají včetně přípony.

#### **Sekce: [Global]**

Popis: Globální parametry

Klíče:

JMÉNO KLÍČE	TYP HODNOTY	POPIS
Problem_type	int	Druh řešené úlohy. V současnosti je podporována hodnota 1 = ustálené saturované proudění, 2 = neustálené proudění a 5 = ustálené proudění pomocí Primaru
Description	string	Krátký popis úlohy - libovolný text

#### **Sekce: [Input]**

Popis: Jména vstupních souborů programu FLOW123D

Klíče:

JMÉNO KLÍČE	TYP HODNOTY	POPIS
Mesh	string	Název souboru popisu sítě
Material	string	Název souboru vlastností materiálu
Boundary	string	Název souboru okrajových podmínek
Neighbouring	string	Název souboru sousedností

#### **Sekce: [Run]**

Popis: Parametry řídící běh programu

Klíče:

JMÉNO KLÍČE	TYP HODNOTY	POPIS
Log_file	string	Název log-souboru.
Screen_verbosity	int	Množství hlášek vypisovaných na obrazovku. (0 = žádné, ..., 7 = všechny)
Log_verbosity	int	Množství hlášek vypisovaných do log-souboru. (0 = žádné, ..., 7 = všechny)

#### **Sekce: [Solver]**

Popis: Parametry řídící chování řešiče soustavy lineárních rovnic.

Klíče:

JMÉNO KLÍČE	TYP HODNOTY	POPIS
-------------	-------------	-------

Solver_name	string	Příkaz spouštějící externí řešič. Podporované hodnoty jsou: "matlab", "gi8", "si2", "gm6"
Solver_accuracy	double	Požadovaná přesnost řešiče - hodnota rezidua, matice, pro kterou se ukončí iterační řešení. Rozumné hodnoty leží mezi 1e-4 až 1e-10. Větší hodnota = rychlejší běh, horší přesnost
Keep_solver_files	YES/NO	Pokud je nastaveno "YES", nejsou smazány vstupní/výstupní soubory řešiče po jeho běhu.
Manual_solver_run	YES/NO	Pokud je nastaveno "YES", je po zápisu vstupních souborů řešiče zastaven běh programu a uživatel musí spustit řešič ručně. Vhodné pro špatně podmíněné či jinak problematické matice.
Use_control_file	YES/NO	Pokud je nastaveno "YES", je jako řídicí soubor řešiče použit externí soubor, jinak je řídicí soubor vytvořen programem PRIMAR/MIXHYB.
Control_file	string	Jméno řídicího souboru, pokud je nastaveno Use_control_file = Yes

#### Sekce: [Output]

Popis: Parametry výstupu.

Klíče:

JMÉNO KLÍČE	TYP HODNOTY	POPIS
Write_output_file	YES/NO	Jestliže nastaveno "YES", je zapisován výstupní soubor.
Ouput_file	string	Název výstupního souboru.
Output_digits	int	Počet platných číslic pro zápis floating-point hodnot ve výstupním souboru.

Pro spuštění výpočtu založeném na primární formulaci (včetně algoritmu pro přidávání násobných uzlů) je tedy nutné mít v ini souboru nastaveno Problem\_type = 5.

Nutno podotknout, že následující popisy souborů jsou jen stručné pro potřebu základního nastavení primárního modelu, v případě potřeby podrobnějších vysvětlení odkazujeme na [2] nebo na [www.geuz.org/gmsh/](http://www.geuz.org/gmsh/).

### Soubor popisu sítě

Popis sítě je soubor typu .MSH používaný programem GMSH, viz. Má dvě části:

**Definice uzlů:** v tomto pořadí je identifikační číslo uzlu a jednoznačné určení polohy ve 3D (ve složkách x,y,z)

**Definice elementů:**

- identifikační číslo elementu,
- typ elementu: 1 = 1D, 2 = 2D(trojúhelník) a 4 = 3D(čtyřstěn)
- počet tagů
- první tag je identifikační číslo materiálu pro daný element (musí souhlasit s id v souboru popisu materiálu)
- druhý tag pomocné rozlišení skupiny elementů v síti pomocí čísel 0-19
- id uzlů definujících element

### Soubor materiálů

Soubor je textový, má řádkovou strukturu, pro zápis čísel platí konvence jazyka C.

Definice vlastností materiálů má strukturu:

```
...
$MTR
n_materials
material_id type coef
...
$ENDMTR
...
```

Veškeré informace před řádkem "\$MTR" a za řádkem "\$ENDMTR" jsou ignorovány.

**n\_materials:** určuje počet definovaných materiálů.

**material\_id:** identifikační číslo materiálu, unikátní pro každý materiál.

**type:** typ materiálu 11 = k (pro 1D element), 21 =  $\begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix}$  (pro 2D element),

31 =  $\begin{pmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & k \end{pmatrix}$  (pro 3D element)

**coef:** pro náš případ hodnota k

### **Soubor okrajových podmínek**

```
$BoundaryFormat
1.0 0 8
$EndBoundaryFormat
$BoundaryConditions
n_boundaries
boundary_num boundary_type boundary_value where node_num
...
$EndBoundaryConditions
```

**boundary\_type:** může nabývat hodnot 1 = Dirichlet, 2 = Neumann

**where:** nabývá v případě výpočtu Primarem pouze 1 = zadáno do uzlu

**node\_num:** je identifikační číslo uzlu, ve kterém je okr. podmínka zadána

### **Soubor sousedících elementů**

Za předpokladu, že je definován soubor popisující síť, materiál a okrajové podmínky, umí Flow123D vytvořit poslední potřebný vstupní soubor – soubor sousedících elementů. Jestliže se odkážeme na část přílohy spuštění programu, jediný rozdíl pro vytvoření tohoto souboru je ten, že místo „-s“ se zadá „-n“.

**Příklad:** flow123d.exe -n kanal.ini

### **Soubor výsledku výpočtů**

Tento soubor má formát .POS souborů programu GMSH. Jeho struktura je popsána v dokumentaci programu (konkrétně se jedná o „parsed-post-procesing file“). V tomto souboru jsou uloženy hodnoty skalární veličiny (tlaku/tlakové výšky) na jednotlivých elementech sítě a hodnoty rychlosti proudění na elementech.